Commodore 64: Comparison of 4 Wordprocessing Packages

The Transactor

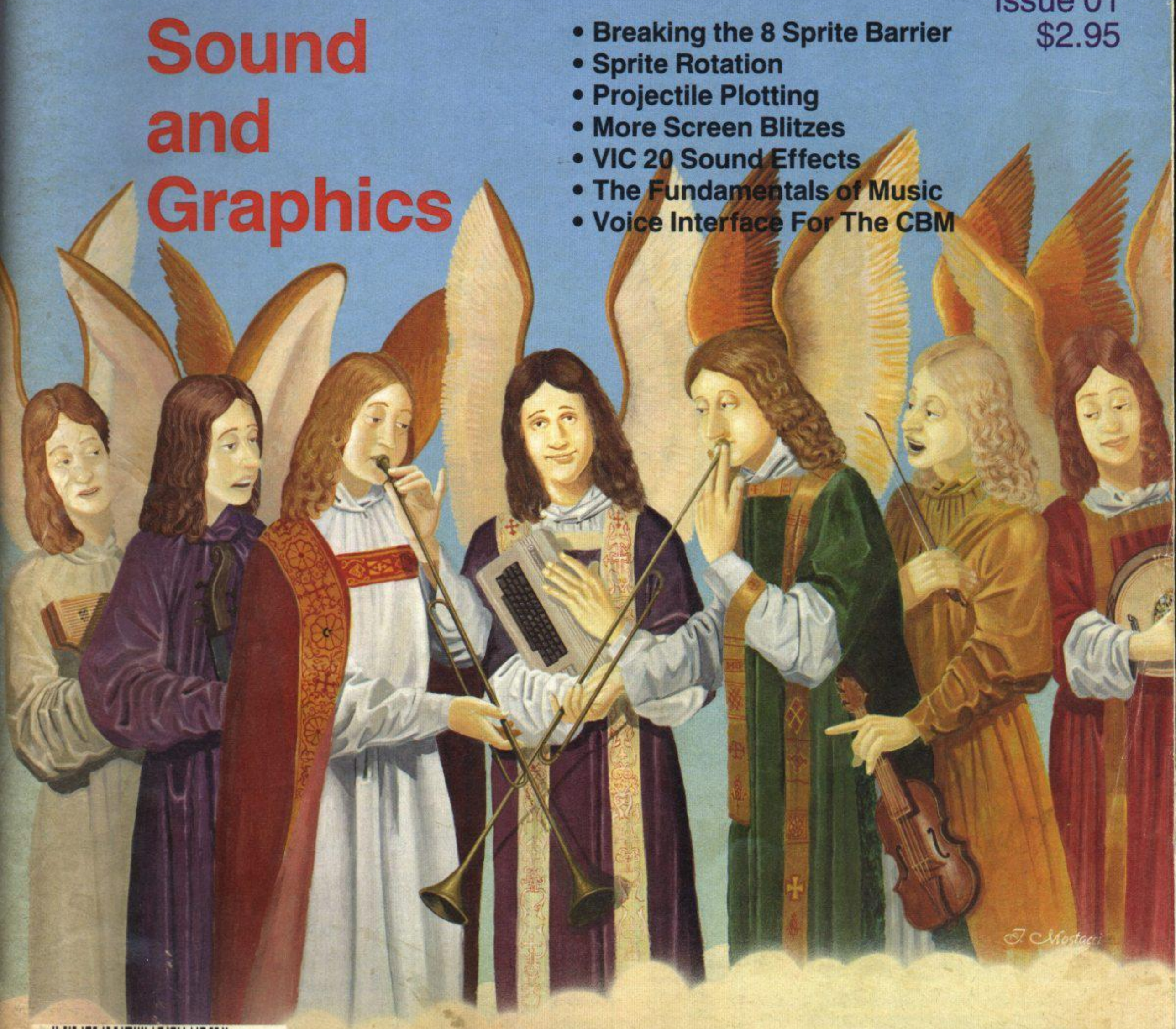
Second Class Mail Permit Pending
Postage Paid in Milton, Ontario

The Tech/News Journal For Commodore Computers Vol. 5.

Issue 01
$2.95

Sound and Graphics

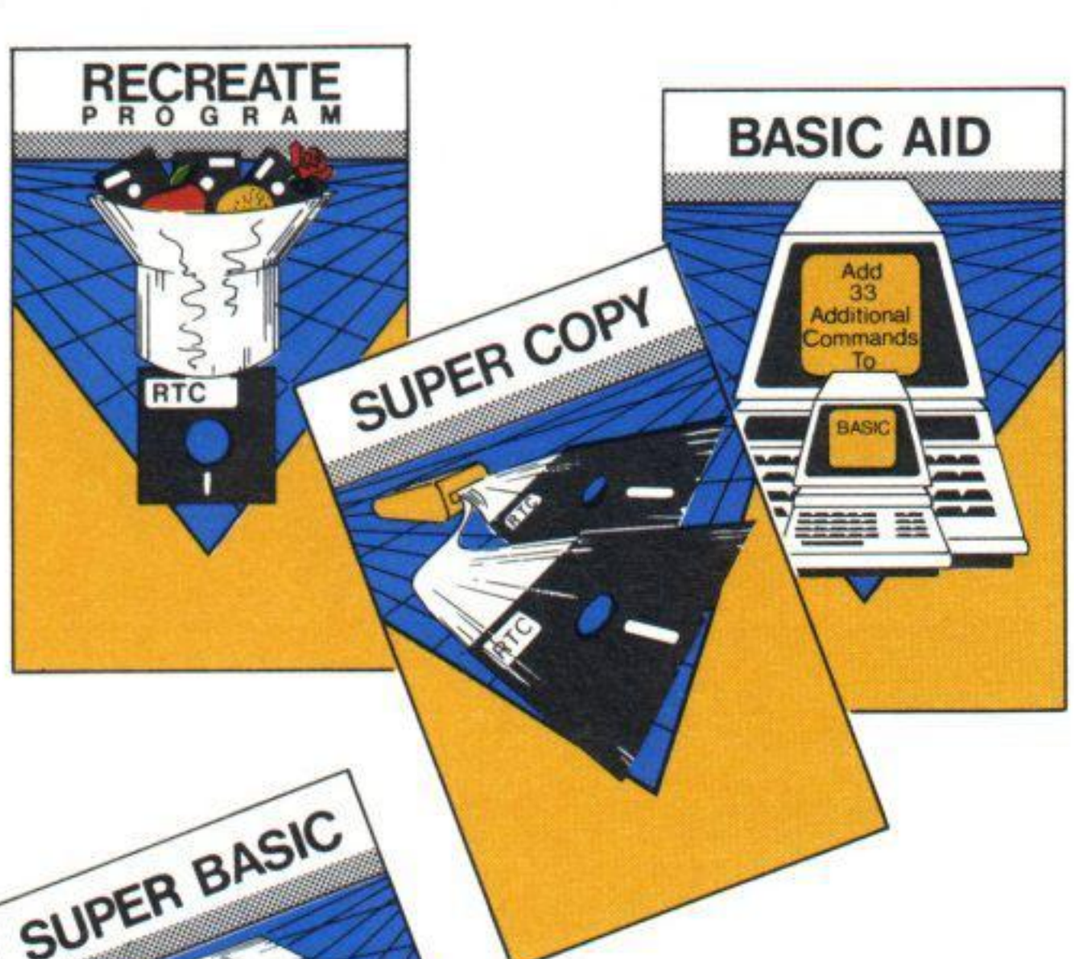
- Breaking the 8 Sprite Barrier
- Sprite Rotation
- Projectile Plotting
- More Screen Blitzes
- VIC 20 Sound Effects
- The Fundamentals of Music
- Voice Interface For The CBM



0 01
57913 70974

Plus: A First Look at the New Commodore 264

Developing a mind for the Future. RTC



Color 80 **$59.95** CDN.

Produces 80 Columns With Color!
Needs No Extra Hardware
Use With Terminal Programs
Uses No Basic Memory in the 64
Develop Your Own Programs

NEW Combined PACKAGE

SCRIPT 64 & SCRATCH PAD 64 **$129.00** CDN.

Script 64:
Word Processor in French and English
80 Columns
Global Search and Replace
User Created Dictionary
Spelling Check
Scratch Pad 64:
The Database/Mail List in One!
Merges with Script 64 Word Processor
Print out Labels, Envelopes, Mail List & More!
Suitable with both Single and Dual Disk Drives
Fully C64 Link Compatible

SUPER BASIC **$46.95** CDN.

- Gives you 3 different versions of Commodore Basic Programing Language Version 4 PLUS!
  A Built in Machine Language Monitor!
- Disk & File Maintenance Commands
  Data Handling Commands
  Graphics Plus Basic
  Compatible with Commmodore's "B" Series &
  Much Much More!

RECREATE PROGRAM **$39.95** CDN.

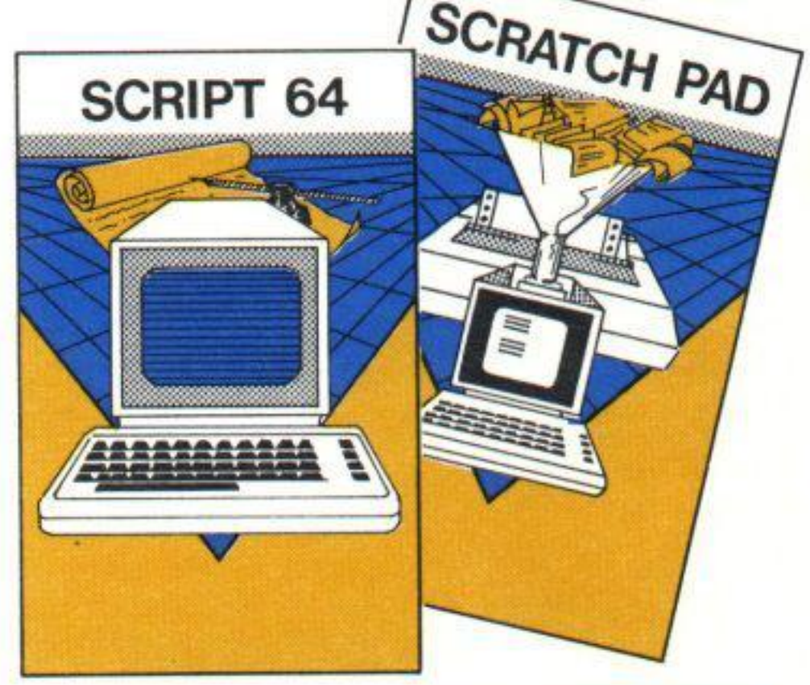
RTC's Answers to Program Recreation
Converts Printer's File to SCRIPT 64's Files

SUPER COPY **$39.95** CDN.

Super Fast Disk Copies on a 1541
Copy Entire Disk in 7 Minutes or Less
Copy Selected Files
Complete Pattern Matching
Full Prompts

BASIC AID **$49.95** CDN.

Your Aid to Writing Programs
Allows Scrolling Through Programs
Adds 33 more Commands to the Basic Language
Has Find, Change, Merge, Move Commands
Convert Hex, Binary and Decimal Numbers and More!



C64 LINK©

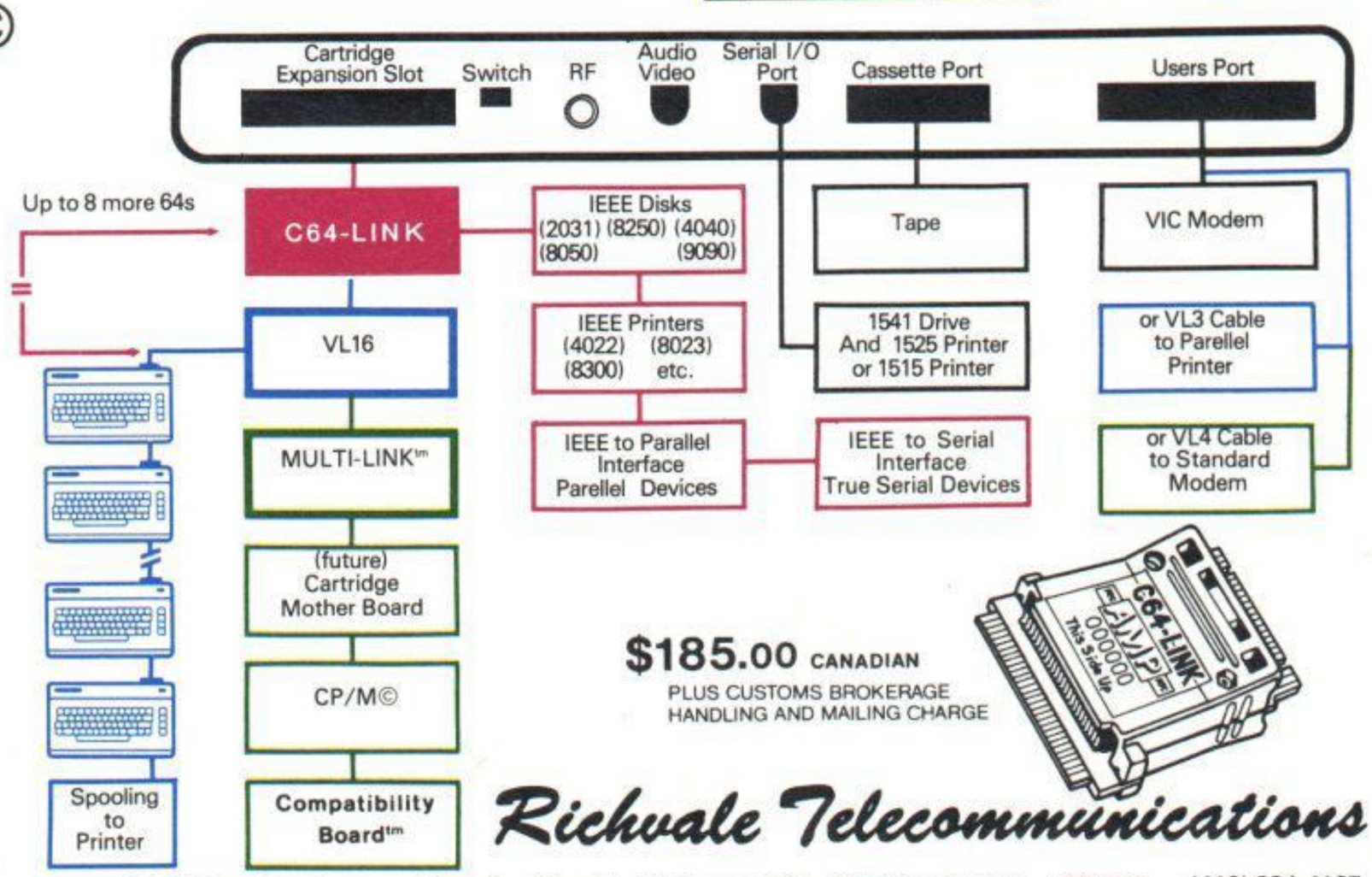
The Smart 64

Give These Expanded Capabilities to Your 64 and VIC 20

* The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)
* BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.
* The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.
* Built-in machine language monitor.
* A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.
* Compatability with CP/M.

Contact your local Commodore dealer or RTC.

Payments by VISA, MASTERCARD or BANK TRANSFER.
Mail orders also by certified cheque, etc.



$185.00 CANADIAN
PLUS CUSTOMS BROKERAGE
HANDLING AND MAILING CHARGE

Richvale Telecommunications

10610 Bayview Avenue (Bayview Plaza) Richmond Hill, Ontario, Canada L4C 3N8 (416) 884-4165

CAD/CAM! DON'T SPEND 25k, 50k or $500,000 BEFORE YOU SPEND $79.00

OBJECTIVES

This book will provide managers, engineers, manufacturing personnel and any interested persons an understanding of the fundamentals of Computer Aided Design (CAD) and Computer Aided manufacturing (CAM) applications and technology.

PROGRAM DESCRIPTION

The program will expose you to the various CAD/CAM terminologies used. Hardware and software comparisons will be explored with heavy emphasis on their advantages and disadvantages. Cost justification and implementation are presented using case studies.

WHO SHOULD PARTICIPATE

The course is designed for but not limited to:

— Those managers, engineers and research professionals associated with the manufacturing industry.

— Personnel from Product, Tool Design, Plant Layout and Plant Engineering who are interested in CAD/CAM.

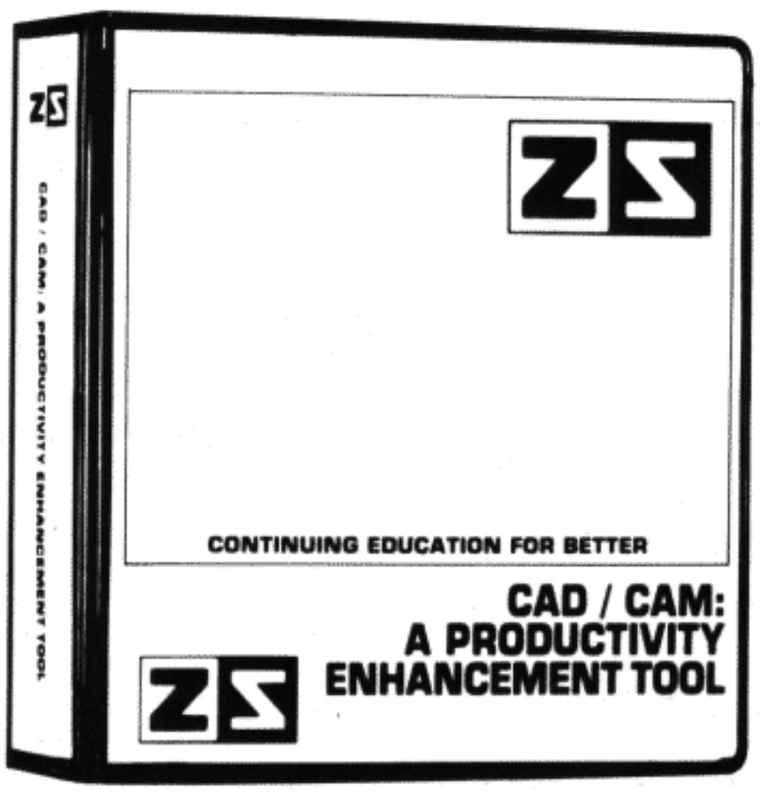
ADVANTAGES— END RESULT

This program will enable participants to:

1. Learn basic CAD/CAM Vocabulary.
2. Better understand the various hardware and software components used in a typical CAD work station.
3. Select the existing CAD/CAM system most appropriate for current and projected needs.
4. Make an effective cost justification as to Why they SHOULD or SHOULD NOT implement a CAD/CAM system.
5. Apply and use computer graphics as a productivity tool.

PROGRAM CONTENT

1. Introduction
   a. History of CAD/CAM
   b. Importance of CAD/CAM
2. Graphics work station peripherals
   a. Input
   b. Output
   c. Advantages and disadvantages of input and output devices.
3. Computer Graphics Systems (Hardware)
   a. Micros
   b. Minis
   c. Main Frames
   d. Turnkey Graphics systems
4. Software
   a. Operating systems
   b. Graphics Packages
   c. Graphics Modules
5. Computer Aided Design
   a. Geometric Definitions (Points, Lines, Circles, ETC..)
   b. Control functions
   c. Graphics Manipulations
   d. Drafting Functions
   e. Filing functions
   f. Applications



6. Implementation
   a. Determining needs
   b. Purchasing and Installing
   c. Getting Started
7. Cost Justification and Survey
   a. Cost comparisons of two and four work station systems.
   b. Presentation of recent survey of CAD system users

ZANIM SYSTEMS MAKES THIS SPECIAL OFFER: IF YOU BUY **CAD/CAM: A PRODUCTIVITY ENHANCEMENT TOOL** BEFORE APRIL 15TH, WE WILL INCLUDE *FREE OF CHARGE* THESE TWO PAPERS PUBLISHED NATIONALLY BY ZANIM SYSTEMS CAD/CAM EXPERT.

1. "Creation of a Large Data Base for a Small Graphics System"
2. "Shortest Path Algorithm Using Computer Graphics"

Of course you could spend as much as $495, $595 or $695 for a similar 3 day seminar even though this book is not a computer program.

We tell you April 15th for a special reason...this product may be tax deductible depending on your field or needs. This 170 page course will satisfy any of your CAD/CAM needs. We guarantee it.

Please send $79 to:

ZANIM SYSTEMS
CAD/CAM GROUP
P.O. BOX 4364
FLINT, MI 48504
(313) 233-5731

QUANTITY DISCOUNTS AVAILABLE FOR COLLEGES, UNIVERSITIES AND/OR SEMINAR USE.

INTRODUCING

THE PRO-LINE TEAM

PAL 64
The fastest and easiest to use assembler for the Commodore 64.+ Pal 64 enables the user to perform assembly language programming using the standard MOS mnemonics. **$69.95**

POWER 64
Is an absolutely indispensible aid to the programmer using Commodore 64 BASIC. Power 64 turbo-charges resident BASIC with dozens of new super useful commands like MERGE, UNDO, TEST and DISK as well as all the old standbys such as RENUM and SEARCH & REPLACE. Includes MorePower 64. **$69.95**

TOOL BOX 64
Is the ultimate programmer's utility package. Includes Pal 64 assembler and Power 64 BASIC soup-up kit all together in one fully integrated and economical package. **$129.95**

SPELLPRO 64
Is an easy to use spelling checker with a standard dictionary expandable to 80,000 words. SpellPro 64 quickly adapts itself to your personal vocabulary and business jargon allowing you to add and delete words to/from the dictionary, edit documents to correct unrecognized words and output lists of unrecognized words to printer or screen. SpellPro 64 was designed to work with the WordPro Series and other wordprocessing programs using the WordPro file format. **$69.95**

MAILPRO, SPELLPRO and PAL are available for Commodore 8032 computers equipped with either Commodore 4040 or 8050 Disk Drives. Further information on request.

WORDPRO 64
This brand new offering from the originators of the WordPro Series brings professional wordprocessing to the Commodore 64 for the first time. Two years under development, WordPro 64 features 100% proportional printing capability as well as 40/80 column display, automatic word wrap, columnization, alternate paging for headers & footers, four way scrolling, extra text area and a brand new 'OOPS' buffer that magically brings back text deleted in error. All you ever dreamed of in a wordprocessor program, WordPro 64 sets a new high standard for the software industry to meet. **$69.95**

MAILPRO 64
A new generation of data organizer and list manager, MailPro 64 is the easiest of all to learn and use. Handles up to 2,000 records on one disk, prints multiple labels across, does minor text editing ie: setting up invoices. Best of all, MailPro 64 resides entirely within memory so you don't have to constantly juggle disks like you must with other data base managers for the Commodore 64. **$69.95**

+Commodore 64 and Commodore are trademarks of Commodore Business Machines Inc.

PRO·LINE SOFTWARE

(416) 273-6350
755 THE QUEENSWAY EAST, UNIT 8,
MISSISSAUGA, ONTARIO, CANADA, L4Y 4C5

The Transactor

Volume 5 Issue 01

Editorial 5
News BRK 6
Letters 16
Bits and Pieces 18
CompuKinks 23
The MANAGER Column 26
Comparison Review:
Four Wordprocessors For The 64 33
The New Commodore Computer: C264 . 43
Subroutine Eliminators 45
A Quick Musical Tour 46
Waves For The 64 49
Programming Sound On The VIC 20 50
Sound Effects 52
The SID In Review 55
Sound Maestro 56
Sprite Rotate 58
QUASIMOB: Break The 8 Sprite Barrier . 60
Changing The C64's Screen Colours ... 64
Simple Harmonic Motion 65
Projectile Motion 68
Voice For Commodore Computers 70
Hardware Corner 72
How Cartridges Work 74
Generator Programs 75

Advertising Section 78
Advertising Index 92

The Transactor
The Tech/News Journal For Commodore Computers

Managing Editor
Karl J. H. Hildon

Editor
Richard Evers

Advertising Manager
Kelly M. George
416 826 1662

Art Director
John Mostacci

Contributing Writers
Eric Armson
Don Bell
Dave Berezowski
Daniel Bingamon
Brad Bjorndahl
Jim Butterfield
Chuan Chee
Domenic Defrancisco
Bob Drake
Jeff Goebel
Melissa Gibbins
Dave Gzik
Fred Hambrecht
Paul Higginbottom
Dave Hook
Mike Panning
Howy Parkins
George Shirinian
Darren J. Spruyt
Brad Templeton
Colin Thompson
Vikash Verma
Don White
James Whitewood
Chris Zamara

Production
Attic Typesetting Ltd.

Printing
Printed in Canada by
MacLean Hunter Printing

The Transactor is published quarterly by Transactor Publishing Inc. It is in no way connected with Commodore Business Machines Ltd. or Commodore Incorporated. Commodore and Commodore product names (PET, CBM, VIC, 64) are registered trademarks of Commodore Inc.

Volume 5 Subscriptions: Canada $15 Cdn
U.S.A. $15 US.
All other $18 US.

Second Class Mail
Permit Pending

Send all subscriptions to: The Transactor, Subscriptions Department, 500 Steeles Avenue, Milton, Ontario, Canada, L9T 3P7, 416 876 4741. From Toronto call 826 1662. Note: Subscriptions are handled at this address ONLY.

The Best of The Transactor Volume 3: $17 Cdn., U.S.A $19 US., all other $21 US. Volume 4 back issues: $4.50 each. Volume 4, Issues 4, 5, & 6 no longer available.

Quantity Orders:

ACCESS
Access Computer Services
630B Magnetic Drive
Downsview, Ontario, M3J 2C4
(416) 736 4402
Dealer Inquiries ONLY:
1 800 268 1238
Subscription related inquiries
are handled ONLY at Milton HQ

CompuLit
PO Box 352
Port Coquitlam, B.C.
V5C 4K6
604 464 3396

U.S.A. Distributor: Prairie News, 2404 West Hirsch, Chicago, IL, 60622, (312) 384 5350

Want to advertise a product or service? Call or write for more information.

Editorial contributions are always welcome and will appear in the issue immediately following receipt. Remuneration is $40 per printed page. Preferred media is 2031, 4040, 8050, or 8250 diskettes with WordPro, WordCraft, Superscript, or SEQ text files. Program listings over 25 lines should be provided on disk or tape. Manuscripts should be typewritten, double spaced, with special characters or formats clearly marked. Photos of authors or equipment, and illustrations will be included with articles depending on quality. Diskettes, tapes and/or photos will be returned on request.

All material accepted becomes the property of The Transactor. All material is copyright by Transactor Publications Inc. Reproduction in any form without permission is in violation of applicable laws. Please re-confirm any permissions granted prior to this notice. Solicited material is accepted on an all rights basis only. Write to the subscriptions address above for a writers package.

The opinions expressed in contributed articles are not necessarily those of The Transactor. Although accuracy is a major objective, The Transactor cannot assume liability for errors in articles or programs.

Program Listings In The Transactor

All programs listed in The Transactor will appear as they would on your screen in Upper/Lower case mode. To clarify two potential character mix-ups, zeroes will appear as '0' and the letter "o" will of course be in lower case. Secondly, the lower case L ('l') has a flat top as opposed to the number 1 which has an angled top.

Many programs will contain reverse video characters that represent cursor movements, colours, or function keys. These will also be shown exactly as they would appear on your screen, but they're listed here for reference.

Occasionally programs will contain lines that show consecutive spaces. Often the number of spaces you insert will not be critical to correct operation of the program. When it is, the required number of spaces will be shown. For example:

print" flush right" – would be shown as – print"[space10]flush right"

Cursor Characters For PET / CBM / VIC / 64

Down – q
Up – Q
Right –]
Left – [Lft]
RVS – r
RVS Off – R
Insert – T
Delete – t
Clear Scrn – S
Home – s
STOP – c

Colour Characters For VIC / 64

Black – P
White – e
Red – £
Cyan – [Cyn]
Purple – [Pur]
Green – ↑
Blue – ←
Yellow – [Yel]
Orange – A
Brown – U
Lt. Red – V
Grey 1 – W
Grey 2 – X
Lt. Green – Y
Lt. Blue – Z
Grey 3 – [Gr3]

Function Keys For VIC / 64

F1 – E
F2 – I
F3 – F
F4 – J
F5 – G
F6 – K
F7 – H
F8 – L

From The Editor's Desk

New Limitations

Volume 5, Issue 01 marks a new boundary crossed - the 90 page barrier. This is our biggest issue yet - quite a change since our humble beginnings as a 2 page newsletter stapled at the top left corner.

That was 1978. Only a year before Commodore released the PET 2001. So, for all intents and purposes, the industry that we've come to know and love as the world of microcomputers is just over seven years old. Seven years. . . not very long as history standards go. The car, the airplane, and the telephone are considered relatively new inventions. Yet they haven't *really* changed much considering they've been around for somewhat longer. Like the computer, these everyday items have changed life on Earth to the point that we would only notice them if they were taken away.

But with one difference – aside from the more esoteric activities, the car, plane and phone service only one fundamental requirement of life. So where does that leave the computer? Micro pragmatics change like the weather. Only seven years old and destined for a future of constant inner turmoil and endless 'version'itis. The changes we've seen will pale by comparison with those no doubt in store. I've heard some say stressfully, "Where will it end?". Well, it's not gonna! Actually it's only beginning!

But so what! What could be more perfect. Certainly it will never get boring. Just as we get to know the latest technology, a new one will arrive. . . with new commands to learn, new manuals to criticize, and new bugs to attend to. And it will continue to do so, faster each time.

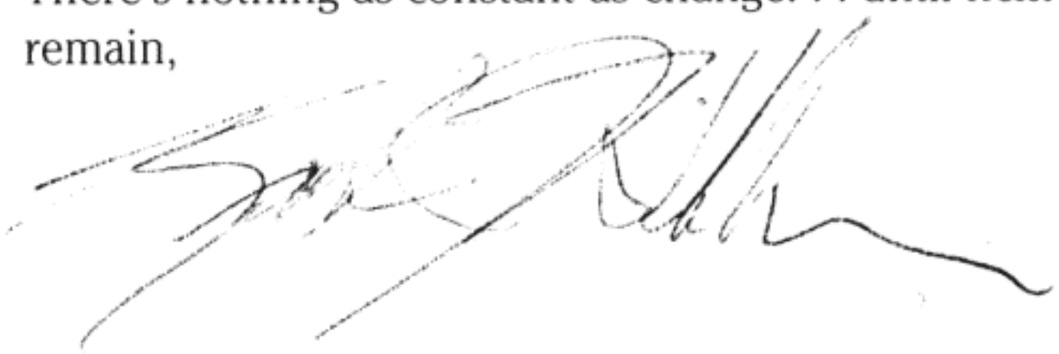
Reports claim that one out of every twenty homes own a microcomputer. The same reports say that by 1990, that ratio will be 10 in 20. Staggering. In just 6 years there will be 10 times as many of us. You think it's overwhelming now – just think how the 10th person will feel.

We have an advantage now: experience. Most of us have seen one system come and go. Even if you're still on your first machine, that experience alone will make the new machines easier to digest. Those entering the field 4 years from now will be faced with mega–features to assimilate. For us it will be a simple matter of learning the next step. And each time there is one, we acquire that much more discipline – the effort required to learn will become less and less. For those who wait, the effort will geometrically increase with each new generation.

But that's assuming we follow the path beaten down by one manufacturer. Herein lies a trap that must be avoided if we are to maintain the edge. Check out the other manufacturers – the neophytes are. Granted you probably did too just before you made that 'decision to buy', but don't turn off now. Keeping abreast of what's out there is just as important as knowing what's in there! Don't be naive enough to think that just because you have mastered machine language on your Commodore 64 makes you a veteran and you can coast for a while. Sure, the next generation will advance along a pattern set by the predecessor. But the other brands will be advancing too, and before you know it you'll be the one with catching up to do. You don't have to know their memory maps inside out – just a grasp of the basic differences and their fundamental operation and applications will keep you in tune, AND gain you respect among your peers.

The microcomputer world has become much more than just learning to program. No, there's much more to it than that. The desire to learn is probably the most important step, and if sticking with it is firmly in your plans you have a good head start. Keep it!

There's nothing as constant as change. . . until next issue, I remain,



Karl J.H. Hildon
Managing Editor, The Transactor

News BRK

Subscribers U.S.A.

Some of you that have sent in the postage paid subscription cards to our U.S. office may have had the cards returned. Our apologies for this error. Please re–send the card as is. This problem has since been cleared up and it won't be returned again.

Commodore News

Jack Tremiel Resigns

President and Founder of Commodore International, Jack Tremiel, has resigned after taking his company to the $1 billion dollar mark in sales. Tremiel started Commodore 25 years ago in Toronto, Ontario as a typewriter repair shop on Yonge Street. Since 1958, Commodore has seen many products come and go – typewriters, radios, LED & LCD watches and calculators, office furniture – and Tremiel has shown a uncanny sense for knowing when to get into a market AND get out.

Reasons for Jacks carreer decision vary from personal to professional conflict. Reportedly, he will act as a consultant for Commodore. Whatever Jacks' future holds, we at The Transactor wish him well and thank him for being a true contributor to the advent of the microcomputer.

Commodore Introduces Productivity Software For Commodore 264 and Commodore 64

Commodore has introduced several productivity software packages for the Commodore 264 and Commodore 64 home computers.

"This is the most powerful assortment of productivity software ever introduced for a personal computer," said Sig Hartmann, president of Commodore Software. "All of these packages will be available by late spring on cartridge or disk for the Commodore 64 and Commodore 264. Several of these packages will be available as BUILT–IN software options for the Commodore 264."

The new products include:

• MAGIC DESK II, a user–friendly program with integrated text editor, spreadsheet, file manager, and calculator for beginning computer users. Special "help screens" are built–in, and instead of commands, MAGIC DESK uses PICTURES of commonly used items.

• Commodore 3–PLUS–1, the first integrated software for a home computer. It includes a word processor, file manager, and spreadsheet. . . PLUS. . . graphics! Commodore 3–PLUS–1 uses windowing to allow the use of the word processor and spreadsheet simultaneously on the screen.

• SUPERSCRIPT 264, a multi–function professional word processor designed for both beginner and expert users. It includes text editing, number calculations, mail list functions, and a cut and paste feature that adds easy on–screen text editing.

• EASYCALC 64 and EASYCALC 264, full–featured spreadsheet programs with color selection and graphics. EASYCALC 64, for the Commodore 64, and EASYCALC 264, for the Commodore 264, are both sold on cartridge, providing more working space within the computer than comparable disk–based spreadsheets.

• Commodore B/GRAPH, an easy–to–use business graphics and statistics package for businesspeople and students. B/GRAPH computes and converts financial and statistics results into colorful 3–dimensional charts, graphs, pie charts, histograms, and other graphics.

• FINANCIAL ADVISOR, a sophisticated financial aid computes loan, mortgage, and investment formulas. It is available on easy–to–use plug–in cartridges.

• TELIGRAPHICS, videotext and graphics software for use with Commodore telecommunications modems. It allows transmission of pictures, text, and business graphics over the telephone and between computers. TELIGRAPHIC also allows users to upload and download data through telecomputing services like CompuServe. Compatible with Canadian Telidon videotext standard.

"As with all of our products, we will market our new productivity software at consumer prices," Hartmann said. "We are pleased that Commodore is able to take the leadership position in raising the quality of software available to personal computer owners." For more information contact Sig Hartmann:

Commodore Computer Systems Division
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

Commodore Introduces 100 Applications Templates for The Manager

Commodore Software today introduced a series of application template products for its Commodore 64 MANAGER 64 database system. The templates include five to ten specific applications provided on each disk, which, when used in conjunction with the MANAGER, allow the user to easily computerize home budgets, index recipes, keep track of sports statistics, track business accounts, and more. Each application comes with complete documentation and built-in help screens.

"The MANAGER is a powerful, sophisticated database program that lets the user define his own applications," said Paul Goheen, director of applications software. "Four applications come with the MANAGER: 'Holiday Planner/Mailing List,' Task Manager,' 'Portfolio Manager,' and 'Home Checkbook.' These applications make it easy for the first-time computer user to immediately take advantage of the capabilities of computerized record keeping."

The ready-to-use templates include the HOME MANAGER, the KITCHEN MANAGER, SPORTS MANAGER, and the BUSINESS MANAGER. One hundred different applications are planned.

Typical of the applications templates is the HOME MANAGER, which provides five different applications:

• Home Inventory—Organized household effects for insurance and business purposes.

• Home Budget—Tracks where your money goes each month. It comes with 15 pre-defined budget categories set up for each month with room for 10 additional categories of your own choosing. A bar graph comparison of budgeted versus actual expenses is also provided.

• Birthday/Mailing List—An electronic address book that remembers what birthday gifts have been given for the past five years and provides a list in calendar date order of birthdays, including actual ages.

• Garden Records—Inventories plants with date of planting and retains up-to-date information on soil mixture, fertilizer used, and weekly crop yields.

• File Card—A computer index card that stores notes, recipes, things to remember, names and addresses, etc. Each card has an optional title entry that makes searching for a specific card easier. Capabilities of the MANAGER include data entry screens, full search and sort capability, arithmetic calculations and a powerful report generator. For more information contact Paul Goheen at the number or address above.

Commodore Successful With Outside Software Development

Commodore Business Machines today introduced a variety of new third party programs which it feels are potential sales hits because of their creativity and originality. Among them are International Soccer, Jack Attack, Micro Illustrator, Micro Cookbook, 3 Plus 1, and the Milliken Edufun! series. These are all products of Commodore's ongoing strong relationship with third party software developers.

In the past, a number of the best selling Commodore software products have been produced by outside developers. These include Easyscript 64, The MANAGER, LOGO, the ZORK adventure series, and the BALLY/MIDWAY game series.

Sig Hartmann, president of Commodore Software states, "One of the major keys to our software success is the independent software developer. We are now working with over 35 of the most creative software developers in the industry." Of the over 200 Commodore branded software products in distribution, over one-half were produced by outside developers. Additionally, over 100 new software products are currently being developed for Commodore by outside software firms such as Data 20, Digital Research, Infocom, InfoDesigns, Island Graphics, MicroPro, Midwest Software, Milliken, and Tri Micro.

"Commodore believes that software acquisition is a two-way street," continues Hartmann. "Independent developers want us to market their software through our worldwide distribution network and we want to make these superior products available to the consumer for the most affordable price. In addition, by tapping the resources of the independent software industry, Commodore is able to obtain the most current, state-of-the-art products available." For more information contact: Sig Hartmann, 215-431-9100

Commodore Announces New Donation Programs for Schools

Commodore has plans for several new donation, grant and information programs for schools and state departments of education, including two special equipment donation programs totaling over $1,000,000.

One such program entitled "CREWS" (Commodore Resources in Education with States) targets the state departments of education. Selection of states is determined by the leadership role taken in assisting local school districts with training, software selection and planning for uses of microcomputers in education. Four states have received equipment donations thus far and many others have been contacted. States receiving equipment include California, New York, Pennsylvania and Texas.

A second donation program offers matching grants to individual schools developing innovative computer education programs in teacher training, curriculum development using applications software, or community education. Any equipment awarded must be "matched-in-kind" by the institution receiving the donation. Interested individuals or schools should write to Commodore Business Machines, Education Department, 1200 Wilson Drive,

West Chester, PA 19380 for information. Submission deadline for requests is March 15, 1984.

Commodore has also developed Education Resource Centers at "Commodore–using" public and private schools and colleges linked with Commodore through a telecommunications network. Each Resource Center agrees to share software, curriculum and teaching strategy information with each other and the communities they serve. In return, Commodore provides each with a free VICMODEM and subscription to CompuServe as well as updated information, including hardware and software products, educational support materials, school computer events, resources for decision making, and technical assistance. According to Dr. David Rosenwald, Commodore's director of educational sales, "The program is growing rapidly and response from schools is very enthusiastic." For more info, contact D. Rosenwald at the Commodore Pennsylvania office.

Commodore & CompuServe Sign Agreement Offering Vidtex

Commodore has signed an agreement for CompuServe's VIDTEX terminal emulator, a telecommunications package that allows users to transfer programs from CompuServe's large library to their own systems for use or disk storage. The agreement allows Commodore to sell, market and distribute VIDTEX worldwide. In addition, CompuServe will also market the package direct and through its dealer network.

In making the announcement, Paul Goheen, director of business software development said, "This is one of the most user–friendly terminal programs ever developed. It is designed to help people who have never used a computer before. By connecting their computer to a telephone via a modem, they can have access to services that provide technical information, free software, an 'electronic magazine,' and 'conversation' between users."

The VIDTEX package uses CompuServe's exclusive "B" Protocol, which allows 100% error detection and correction of files being transmitted to or from CompuServe, ensuring that the program will work when transferred. VIDTEX even notifies the user if an incompatible program written for another computer is trying to be downloaded. Other features include:

- A complete 32K RAM buffer which can capture data from a host system for immediate use or for disk storage for later use.
- Printer support—using the RAM buffer, users can capture data at 120 characters per second for print–out later on a slower printer.
- 10 programmable function keys which can be loaded with such things as user ID for the service or any series of commands. This gives the user access to a series of commands by pressing one key. By saving these definitions on disk, the user may load and use many different function key configurations again and again.
- Colour graphics and cursor positioning.

VIDTEX is a disk–based program designed for Commodore computers that use the Commodore VICMODEM Model 1600 or the AUTOMODEM Model 1650. These include the C–64, CBM 8032 and the new Commodore generation of home and business computers, the C–264. For more information contact Mike Tomczyk at Commodore HQ in PA.

Commodore Introduces Speech Module For The 64

Commodore has developed true–to–life speech for the Commodore 64 series of computers. The speech module plugs directly into the user port of the Commodore 64. It contains an additional port into which other "talking" and "non–talking" cartridges can be inserted.

The Commodore Speech Module contains a built–in vocabulary of 235 words in a pleasant female voice. The voice speed can be user defined to slow, normal, or fast. The words can be programmed directly from Basic and/or assembler. The user can program music, graphics, and speech simultaneously. The Speech Module supports a separate audio out so that the user may connect the speech output directly to a hi–fi system, a television, or a colour monitor. More words and different voices (male, cartoon characters, etc.) will soon be available on disk and cartridge.

Future educational applications on disk and/or cartridge include the alphabet, counting, spelling, and animals. Higher level applications will include interactive foreign language modules, higher mathematics, and science. Programs will be available from Commodore and third party producers.

Because the Commodore Speech Module can be made to produce any voice and a wide range of sound effects, game cartridges will take on an even more realistic effect. Two released games that currently work with the Speech Module are WIZARD OF WOR and GORF. More games are being prepared for release soon. The Speech Module can support game cartridges of up to 128K bytes.

Commodore's Speech Module has a suggested retail price of $59.95 US. and can also plug directly into Commodore's new portable computer, the SX–64, making it the only "talking" portable on the market. For more information contact Myrddin Jones at Commodore HQ.

Commodore Introduces New Line Of Advanced Entertainment Programs

"Video games have changed dramatically since they were first introduced. Todays's game planers think in terms of 'computer games,' a blend of three–dimensional cartoon animation with sophisticated challenges only a computer can achieve."

With those words, Commodore Software President Sig Hartmann introduced INTERNATIONAL SOCCER, the first in a series of "Gold Medallion" games. Continuing, he added, "This designation is reserved for a special category of elite new game and adventure products that have advanced animated graphics as well as play action that uses thought and strategy."

INTERNATIONAL SOCCER offers three–dimensional animation, realistic perspectives and authentic soccer play action for one or two players. Developed for Commodore's line of home computers, INTERNATIONAL SOCCER has a suggested retail price of $34.95, with initial deliveries expected in January 1984.

Future "Gold Medallion" releases are to include a professional–level basketball game as well as "intellectual games." Other entertainment programs introduced include:

- VIDUZZLES—a series of video puzzles for children.
- JACK ATTACK—an animated strategy game with many variations and levels of play.
- SOLAR FOX—one of the best of the Bally Midway adventure games.

All three of these programs will be available for initial deliveries by spring. For more info contact John Mathias at CBM HQ.

Events

Closing The Gap

A national conference to examine the impact of microcomputer technology for the handicapped will be held in Minneapolis, MN September 13–16, 1984.

Titled "Computer Technology For The Handicapped," the conference will be held at the Raddision South Hotel. Sponsors are CLOSING THE GAP (an international newspaper covering microcomputer applications for the handicapped) and TAM (Technology and Media, forming as a division of the Council For Exceptional Children) in cooperation with Dr. Gillbert Schiffman, Department of Education, Johns Hopkins University and Dr. William Buchanan, Applied Physics Laboritory, Johns Hopkins University.

This conference and its program will bring together special education, rehabilitation and medical professionals from around the world to share advances made by technology to aid the handicapped. In addition, this offering is designed to provide information to parents of handicapped children and disabled individuals, demonstrating the opportunities, advances and support this technology can bring to this population.

Over 80 presentations and three, three–hour workshops are available to all participants. In addition there are six pre–conference workshops featuring beginner to advanced training in the use of microcomputers for the handicapped.

Total cost of the conference is $150 if registration is received before September 1st or $175 for registration after that date. TAM and Group discounts are also available. Additional information may be obtained from:

Paula Barnhart, Conference Manager
CLOSING THE GAP
PO Box 68, Henderson, MN, 56044
(612) 665–6573 or Metro Mpls./St. Paul – 341–8299.

Books

CBBS Directory

The National CBBS Directory is now available to all computer users who communicate over the telephone. This directory contains over 1000 computer bulletin board telephone numbers which are conveniently organized in numeric sequence.

The directory also contains a unique key field which identifies relevant information such as the BBS type, its baud rate, operating hours, and special comments specific to each BBS.

Many of the bulletin boards contain excellent programs which you can download into your system at no charge to you. Provides you with a comprehensive list of national BBS telephone numbers. (SYSOPS: send info for next printing of the Directory) The directory is promptly shipped to you by sending only $2.00 postpaid to:

Thomas Wnorowski
3352 Chelsea Circle
Ann Arbor, MI 48104

Microcomputing Periodicals Directory

Microcomputing Periodicals: An Annotated Directory, the original source of information on microcomputer magazines, is about to release its 10th revised and expanded edition. This unique directory to over 800 magazines, newsletters and newspapers related to the use of microcomputers has been called ". . .very impressive, one of the most meticulous bibliographies I have ever seen, and certainly on a subject of considerable current interest" (Peter Gellatly, Serials Librarian).

Each entry includes title, address, frequency of issue, the year the periodical began publishing, its cost, plus a brief annotation describing the periodical and giving an idea of its scope and content. Also included are a list of indexing and abstracting services to these publications, an appendix of periodicals which have changed their name or ceased publication, and indexes by subject and by country of origin.

This book will be invaluable to consultants, researchers, librarians and anyone else who needs to locate up–to–date sources of information on microcomputers and their applications.

In addition to specific brands of equipment and software and a variety of programming languages, subjects that have publications devoted to them include: accounting, agriculture, business, dentistry, education, engineering, finance, games, genealogy, graphics, the handicapped, law, libraries, marketing, medicine, networks, pharmacy, political science, psychiatry, publishing, religion, speech synthesis, word processing, and many, many others.

The cost is $19.95, including postage (add $2.00 outside N. America).

Microcomputing Periodicals
53 Fraserwood Ave. #2
Toronto, Canada
M6B 2N6

Commodore 64 User's Encyclopedia Now Being Shipped

The Book Company (a division of Arrays, Inc.), a leading publisher of home and business reference books for microcomputer users, announced the shipment of their new publication, "The Commodore 64 User's Encyclopedia."

"With a suggested retail of $14.95, 'The Commodore 64 User's Encyclopedia' is an invaluable resource for any Commodore 64 owner. The information on consumer products alone is worth many times the price of purchase. Moreover, as a convenient reference for all the problems and decisions involved in personal computing, it will save buyers hundreds of hours of time that would otherwise be wasted looking elsewhere for necessary information," commented Hank Schienberg, executive vice president of Arrays, Inc.

"The Commodore 64 User's Encyclopedia" addresses the beginning to intermediate user of this best–selling home computer. It is not only especially helpful to owners using the Commodore as a personal or business computer, but it is also very helpful to those making a transition from playing games to exploring other possibilities in personal computing.

"The Commodore 64 User's Encyclopedia" includes all the information a user needs to successfully program in Assembly language and BASIC: sample program lines and clear examples of correct

entry formats; a full description of all operating system commands and functions, including error messages and appropriate responses; explanations of peripheral devices, their functions and operation; definitive information about all commercially available software and hardware enhancements for the Commodore 64, including the most thorough, up–to–date vendors list available; and complete information about further resources for owners of the Commodore 64: books, magazines, user's groups, and other sources of support and education.

The Book Company's user encyclopedia can be purchased from leading bookstores and computer stores across the country. For additional information, contact:

Linda Feldman
The Book Company
11223 South Hindry Avenue,
Los Angeles, CA
90045 (213) 410–9466

Canadian Directory of Software HOTLINE Service

Searching for Software? We can help you find it!

Our database has over 2500 Canadian, American and Europian packages listed. This large selection of packages gives you the chance to find the package best suited to your needs. The database is updated daily with newly evaluated packages.

Simply specify:
1. The Application. . .eg. accounting
2. The Industry. . .. . .eg. manufacturing
3. The Hardware. . .. . .eg. micro

And for just $35, we will search our database for all the packages fitting your set parameters, and send you all the reports on file by mail as soon as possible. Call or write:

Canadian Directory of Software
Maclean Hunter Bldg.
777 Bay Street
Toronto, Ontario
M5W 1A7 (416) 596 5986

Software

Personal Finance Planning and Analysis

Xana Data Systems of Calgary, Alberta has announced the release of a revolutionary new program, the "Personal Finance Forecaster." Designed for home budget planning and administration, the program combines an analysis of the individual spending patterns of the user, along with his or her current financial situation, to project a budget twelve to thirty–six months into the future. It permits detailed "what if" analysis to aid in planning future spending. Available for the Commodore 64 and disk drive, the package includes a master diskette, a working copy diskette, a personal security key and a comprehensive users manual. The program retails for $99.95 CDN and $79.95 US. For more information contact:

Xana Data Systems
3427 – 12th Street N.E.
Calgary, Alberta
T2E 6S6
(403) 276–6834 Canada
(213) 410–9884 United States

Data*Max For The Commodore 64

DATA*MAX is a flexible, easy to use database system which can balance your checkbook, keep a household budget, record collections, and take care of almost any home filing chore. The software and extensive user guide are written especially for the newcomer to computers. All options are screen prompted and require a single key stroke to execute, eliminating the need for constant referrals to the manual.

DATA*MAX can maintain an unlimited number of 10,000 character files containing up to 200 eight field records each. A sub–file option allows the user to split a full file into two or more smaller ones, enabling expansion beyond the 200 record limit. A special "Escape Key" allows any program option to be aborted without loss or change of data. Use of color displays and full sound prompting give instant visual and aural response to user input and error conditions.

Records can be searched and sorted by ANY field or combination of fields. Special "number crunching" features allow sorting in numerical order and provide a summary of field totals and averages. DATA*MAX can also create special sorted files which can be read by word processors for easy generation of form letters, labels, invoices, etc.

Disk commands such as "Directory" and "Format disk" are available without leaving the program. Commodore and non–Commodore interfaces and printers are supported. Printed reports feature pre–set format and automatic paging. DATA*MAX also allows any report screen(s) or a disk directory to be copied to the printer at the touch of a key. Available only for the Commodore 64 on diskette for $29.95 from:

COMPUTERSTUFF
308 1/2 Green St.
Yankton, SD
57078 605 665 2833.

PractiCalc 64

PractiCalc 64 is a complete computer spreadsheet. Why? Because it combines the sophistication of a large business computer with the ease and affordability of a home computer!

PractiCalc 64 is fully equipped with 22 mathematical functions. It will perform all BASIC operations (+, –, x, ./., exponentiation, logarithm, integer, etc.) as well as trigonometric functions like sine, cosine, and tangent. Row/column functions which work on a range of numbers are also included; maximum, minimum, count, average, etc. PractiCalc allows all of these functions to be combined in formulas for easy calculation.

With the maximum number of rows 250 and columns 100, PractiCalc 64 will handle spreadsheets of up to 2000 cells. Moving around large spreadsheets is simplified by a "GO TO" option which allows the user to select a destination cell.

PractiCalc 64 will also sort information both numerically and alphabetically. To find an entry you placed at the far end of the spreadsheet, PractiCalc 64 has a SEARCH function which will automatically seek formulas, numbers, words, or variations of words (with its wild card option). And finally, PractiCalc 64 has the

ability to represent numbers in bar graphs to see how they compare. The bar graphs, available in high and low resolution, are easily printed to a compatible printer for a 'hard copy' of your spreadsheet.

What was once only available on mainframes is now possible on your Commodore 64 with the complete spreadsheet – PractiCalc 64.

Access Computer Services
630B Magnetic Drive
Downsview, Ont.
M3J 2C4 416 736 4402

ExpandaFox 80 For The C64

(for the Quick Brown Fox word processor Commodore 64 – disk version)

ExpandaFox 80 adds 80 column capability (through software) to your Quick Brown Fox word processor. You now have the ability to display 80 characters across your screen (monitor or television) without the purchase of any additional hardware.

ExpandaFox 80 allows you to display your text as it will appear when printed, completely formated.

Quick Brown Fox
536 Broadway
New York, New York
10012 212 925 8290

QuickMail
(a mailing list & mail merge program for the Commodore 64 – disk version)

QuickMail is an efficient, fast and simple to use mailing list/merge program for the Commodore 64 computer, compatible with the Quick Brown Fox friendly word processor. It's an easy way to keep mailing lists tidy and up to date. QuickMail can even be used alone to generate mailing labels.

QuckMail is so powerful that it integrates names and addresses with the BoilerPlate feature of Quick Brown Fox word processor to produce letter perfect, personalized form letters. It even provides a salutation field. And, QuickMail can add, edit and delete any records.

QuickMail will also generate 600 names and addresses per file with provisions for four open fields (additional information of your choice) for identification and coding. It can sort alpha/numerically (by name or zip code), and search for duplicate names and addresses.

QuickLink
(a smart telecommunications program for the Commodore 64 – disk version)

QuickLink is a smart terminal stand alone telecommunications program. QuickLink works with the Quick Brown Fox word processing program and your Commodore 64 to access and edit information via telecommunications services such as The Source, CompuServe, and Dow Jones Services. This means with Quick-Link you have a powerful, smart, terminal program that will put today's most exciting databases at your fingertips. With it, you have the power to acquire information over the telephone, create files which you can edit, and transmit your own electronic messages to other micros using other word processing programs.

- Downloads and uploads files in sequential, program or new IMG format
- Easy to use, totally disk–based system with a menu driven format
- Easily customized to work with any video text service or private bulletin board
- Works with all serial bus printers and properly interfaced centronics and IEEE printers
- Prints or extracts segments of files to create new file ON LINE of OFF LINE
- Its memory buffer can be set to automatically dump download files to disk
- Can be used with both Commodore's 1600 VICMODEM or 1650 AUTOMODEM. QuickLink can repeatedly dial a list of numbers if used with 1650 AUTOMODEM.
- Interfaces with the 80 column Video Pak cartridge from Data 20
- Accesses C64 DOS WEDGE commands without leaving the program
- Has four user defined keys that allow the user to create keys for automatically loading user–ID and passwords
- Built in timer and alarm

QuickFix
(a Super Debugger for the Commodore 64, Atari 400 and 800, and Apple and Apple IIe – disk version only)

Get the bugs out the first time. With QuickFix you can monitor the effects of your assembly language programming with ease and make sure your computer utilizes your instructions.

Super–Debugger
If you write or are learning to write in 6502 Assembly Language, you may find that debugging your programs can be a tedious, difficult, and time–consuming operation. Even a good machine–language monitor with single–stepping capabilities cannot give you a way to observe exactly how each step of your program effects every register and memory location that your program uses.

SuperDebugger is an innovative single–stepping program which allows you to specify which areas of memory your program utilizes. Between each instruction of your program that you execute, the debugger will display the contents of all the registers of the 6502 as well as the contents of the memory locations that your program is using. In this way you can easily keep track of the state of all the important memories between each instruction.

The commands supplied by Super–Debugger are the following:

CR	– Execute the next instruction
G n	– Set the Program Counter to location n
P n	– Same as G
A n	– Set the Accumulator to n
X n	– Set the X Register to n
Y n	– Set the Y Register to n
J	– Single–step the next subroutine call
B	– Exit debugger
M a, b	– Display the block of memory b bytes long starting with loc.
D	– Stop displaying the last block of memory entered
E	– Erase all the blocks of memory indicated so far
S a;x,y,..	– Store the value x, y,. . . in the memories starting with location a

Quick BASIC Aid

(a BASIC programming tool for the Commodore 64 users. So if you want to learn BASIC, or if you want to improve your BASIC programming skills, Quick BASIC Aid is a program designed to really help you.

Quick BASIC Aid offers advanced programming aids not found in Commodore, Microsoft 4.0 BASIC. These aids are designed by programmers to simplify advanced BASIC programming.

You will never have to write another utility program again! Quick BASIC Aid is menu driven, and supports serial bus and IEEE printers.

Although Quick BASIC Aid is an extensive 36K program, it does not use up any of the user's memory for programming since it works with the user's program from the disk. Therefore, the user's program can be any size since it is never in the computer's memory at the same time.

The Utilities:
- RENUMBER – renumber all, or specify line parameters
- EXTRACT– Create a new subroutine, or program from any BASIC program and merges it with another program. Other features included are MERGE, SEARCH AND REPLACE.
- CROSS REFERENCE – produce a complete cross reference of all variables and line numbers.

Contact Quick Brown Fox for more details.

C64 Master Composer

Master Composer is the perfect utility for programming music on the Commodore 64. It is fun, easy to use and very powerful. Master Composer takes full advantage of the sound synthesizer to produce all types of music from simple melodies to intricate compositions. With Master Composer both the beginner and the accomplished musician will have the power of the Commodore 64 synthesizer at their finger tips. You can compose your own musical scores, experiment with different arrangements, and instruments, program your own accompaniment, or just type in your favorite sheet music.

Music files are easily added to your basic or M/L program and the interrupt driven song plays while your program runs. You may relocate and link files and selectively play different parts of your songs.

ACCESS SOFTWARE INC
925 East 900 South
Sales Lake City, Utah
84105 801 532 1134

Flight Simulator II For The C64

Description

Let Flight Simulator II put you at the controls of a modern single–engine aircraft with full flight instrumentation and colorfully realistic panoramic view. Sophisticated programming techniques, combined with documentation that's fun as well as informative, now make the once–expensive American dream of flying a reality for anyone with an appropriate home computer. Flight Simulator II will train you in everything from basic aircraft control through complex aerobatic flight maneuvers. And when you think your're ready, you can test your flying skills with the "World War I Ace" aerial battle game.

Features

Flight Simulator II simulates the instruments and flight characteristics of a Piper PA–28–181 Archer II, a single engine, 148 mph, non–retractable gear aircraft equipped with a good set of avionics. Extensive flight controls (accessible using either keyboard or joystick) and instrumentation appropriate to visual and instrument flight are included. This aircraft was chosen because it offers good performance yet is simple and easy to fly.

3D Out–the–Window Display

High speed color–filled 3D graphics (2 to 6 frames per second) are used to provide a spectacular view of the outside world. Ground terrain including mountains and water, runways, prominent features, and the horizon make even simple scenic flights a beautiful experience.

Visual effects are very similar to those encountered in real flying. Going through clouds causes the view to become completely white. On clear days the sky is blue. Cloudy days result in a grey sky unless you break through the cloud layer to blue sky.

Airports

Flight Simulator II lets you practice takeoffs and landings, in–flight maneuvers, even complicated aerobatics over realistic scenery. The program features over 80 airports in four scenery area; New York, Chicago, Seattle, and Los Angeles, with additional scenery areas available separately. The potential flying environment (ten–thousand by ten–thousand miles square) encompasses the entire continental United States, with a resolution finer than 2.5 inches.

Instrumentation

Realistic flight instruments and sophisticated avionics (radios) are provided. Ground navigation facilities and aeronautical charts allow you to learn cross–county flying techniques as well as local arrival and departure procedures. Avionics include two VOR receivers (the most commonly–used navigation radios), DME (distance measuring equipment), ADF (automatic direction finder), and complete ILS (instrument landing systems) receivers.

Environmental Factors

Flight Simulator II simulates quite a few of the external environmental conditions that affect flying the most. Season, time of day (day, dusk, or night), cloud layers, winds, and turbulence are all user–adjustable.

Documentation

Two separate manuals are included with Flight Simulator II to help you get the most out of the simulation. The 88 page Pilot's Operating Handbook and 96 page Flight Manual explains aircraft and flight instrumentation controls, editor use, and program features.

The Flight Physics and Aircraft Control manual was developed specifically for users with no previous aviation experience. The manual begins with the theory and physics of aircraft flight. Eight beginning flight lessons then train the user in different aspects of aircraft control plus practical applications of the sophisticated avionics. Finally, a section on aerobatics teaches you how to perform loops, spins, rolls, and other fun aerial maneuvers.

World War I Ace

The program also features a "World War I Ace" aerial battle game that lets you test your flying skills against six deadly enemy fighter

planes. Your mission is to bomb the enemy fuel depots and factories. Six computer–controlled fighters assigned to protect the enemy installations will attempt to engage you in dog–fights when you enter hostile territory. An attack radar screen, machine gun, and five bombs are provided to help you accomplish your mission.

Flight Simulator II is available on disk for the Apple II, Atari, and Commodore 64 computers. A (limited–feature) cassette version may also be available for your system; please call for further information.

If your dealer doesn't carry Flight Simulator II for your computer system, you may have him order it or you may order directly from SubLOGIC. For direct orders plase specify your system and media requirements, enclose $49.95 plus $1.90 for shipping, and specify whether you want UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club charges accepted.

SubLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
(217) 359–8482 Telex 206995

3–D Space Adventure

STAR BATTLE, a realistic space adventure to save Earth, is a new strategy game for Commodore 64 users.

Seated in the command center of the Starship Columbia as its captain, the player attempts to save Earth from the infamous Scyons. The mission takes place in a realistic 3–dimensional galaxy with 64 quadrants. The starship is equipped with warp and hyper–light driver, phasers, photon torpedoes, an advanced tracking/viewing system and OBNAC 3000, the most sophisticated on–board computer in the Star Fleet.

OBNAC stands for Objective Based Navigation and Attack Computer. It helps track down and destroy enemy ships. A keyboard overlay is included to transform the computer into the OBNAC control console.

A player must use a shield energy to protect the Columbia from Scyon Battle Cruisers because its capabilities are reduced each time it is hit by enemy fire. The Columbia captain can return to a Starbase for repairs if the ship is damaged. Can the captain stop the enemy ships, or will the Scyons plunder Earth and reduce the Columbia to Ion dust?

STAR BATTLE is software of substance. It is a strategy game with 3–D graphics, with ten skill levels that will challenge players from 6 to 18 and beyond.

STAR BATTLE is one of TIMEWORKS strategy games, including the very popular ROBBERS OF THE LOST TOMB. Each of the Entertainment games has a suggested retail price of $34.95. Look for these programs at your local computer store, or for dealer information, contact:

Micro Marketing Canada
169 Inglewood Drive
Toronto, Ont.

COMAL 64

COMAL Users Group (USA) has announced the immediate availability of the language for the Commodore 64 designed to replace BASIC. COMAL includes 40 graphics statements and commands, 11 sprite statements, and turtle graphics. It includes a RUN–TIME COMPILER which explains why it is so much faster than BASIC. String searches are actually over 79 times faster, and typical program execution should be about 6 times faster.

It is much easier to program in COMAL since it includes virtually all of the BASIC keywords, plus to program structures popularized by Pascal. When COMAL lists a program, these structures are automatically indented. Variable names can be up to 16 characters long, all characters significant. Procedures and functions are named and can be multi–line. Programming errors are caught as you enter the line, with understandable error messages, and the cursor placed on the line at the point of error. The error message is removed from the screen after you correct the error, and the previous screen line is replaced (non–destructive error messages).

COMAL allows you to store program segments on disk. They can be merged into another program later. Auto line numbering and line renumbering is standard with COMAL, as is End Of Data and End Of File detection. The function keys are used to easily switch from graphics screen to text screen and back. A simple statement can be used to disable/enable the STOP key. And COMAL can read data files created by BASIC, no conversion needed.

The total cost for this new, well designed, powerful language is only $19.95, less than you might expect to pay just for a few "patches" to BASIC. Send a SASE for more information or order the COMAL System Disk from the:

COMAL Users Group
5501 Groveland Terrace
Madison, WI 53716

C64 Physics Lab

(a physics tutorial game for the Commodore 64 – disk version)

If you are a physics student or just curious about how things work, Physics Lab is a program which was designed with you in mind. A joystick and a desire to learn is all that's required.

This program provides a way to gain an intuitive understanding of elementary physics. Its features allow it to be used as a simple game or as a powerful self teaching aid.

Physics Lab provides a simulated real world that includes:

- masses
- uniform gravity
- inclined planes
- springs
- pendulums
- inverse–square gravity
- friction
- rolling

Using a joystick to both make selections and position objects, you can. . .

- put together complicated systems out of elementary items
- observe and record the movement of objects in the system you create
- observe, through the use of advance graphics, the magnitude and direction of the forces that objects exert on each other as they move
- use of joystick to exert external force on masses in the system
- display and superimpose graphs of how the position and velocity of masses vary with time in different experiments
- play a game in which you must build a Rube Goldberg–type mousetrap using the various items of the simulated world and try

to fine–tune its operation with a minimum number of adjustments • learn basic mechanics by observing the way forces in the simulated world balance each other, accelerate masses, act during collisions, and vary between different experiments.

Quick Brown Fox
536 Broadway
New York, New York
10012 212 925 8290

QuickFinger
(a typing tutorial and educational game for the Commodore 64 – disk version)

Let your fingers do the stalking with QuickFinger, an educational game that teaches kids of all ages and adults how to use a computer keyboard. QuickFinger is ideal for home and classroom use; it includes three action–oriented games that combine color and sound to provide an exciting entry to keyboarding without tedious note exercises.

In ALPHA–ZAP, random letters drop out of the sky. The player must prevent these letters from "zapping" a hapless pedestrian strolling across the screen. The letters begin to come faster and faster and before you know it, you're a master on the keyboard.

In KEYBOARD PACER, the standard QWERTY typewriter keyboard is displayed on the screen. At first, letters slowly begin to flash one at a time. Speed and accuracy are tested by players as they attempt to type letters that flash on the keyboard.

In LINK THE LETTERS, the trick is to find three and four–letter words hidden in a maze. This is a good vocabulary building game as well as an innovative keyboard exercise.

From the makers of the popular Quick Brown Fox word–processing program, QuickFinger is your entree to any computer application that requires ease and familiarity with the keyboard. Contact QBF above for info.

Better Letter & Better Writer For The 64

We write your letters. Invitations. Resumes. Announcements. Applications. Greetings. And many more. 100 letters just waiting for you to fill in a few simple details, and. . .PRINT! Select a letter from the disk and load it into your favorite word processor. If you don't have one, use our simple to operate Better Writer supplied with the package. Supports most word processors using your computer's ASCII format. Better Letter & Writer comes on disk for $34.95 US.

Quick Touch Typing For The 64
We teach you typing. Step by step. Easy to follow lessons even kids will love. And then, we make you type FAST. You actually watch your speed increase with every drill. $24.95 US. on disk, $22.95 on tape.

Ez/Disk Plus For The 64
This easy to use program turns your disk drive into a powerful machine! No more lengthy hard to remember BASIC commands. . .totally transparent. . .your program in memory remains intact until Ez/Disk Plus LOADs or RUNs another. Indispensable tool for every drive owner. Ez/Disk Plus features:

• Support for up to four disk drives
• Duplicate entire diskette using one or two drives
• Print any disk file to printer or the screen
• Disassemble machine language files directly from disk
• List disk directory to printer or the screen
• Execute any BASIC or machine language program from disk
• LOAD/B.LOAD any BASIC or machine language program from disk
• SAVE/B.SAVE any BASIC or machine language program from disk
• Append BASIC programs to a memory resident program
• Read drive errors instantly
• Change drive numbers to add more drives
• Rename, Erase and Compress disk files by answering a prompt
• Format or clear new disks
• Quit to BASIC to continue working on your program
• And more. . .

Call Ez/Disk Plus any time by pressing the @ key (and RETURN). The menu appears instantly ready for your commands! $34.95 US. on disk.

Add $2 for shipping. Add $3 for COD. NY residents add 8.25% tax. For more on these 3 packages:

SOFTRON, Inc.
150 Nassau St. Suite 2024
New York, NY
10038 212 608 2922.

Hardware

C64 LASERSOFT

An Interactive Video Disk/Microcomputer Instructional System for the Low–End Market. We call it LaserSoft, and we developed it for the low–end market serving schools and families.

Initially, it has been designed to work with the following equipment configuration: Commodore 64 microcomputer with 1541 disk drive, a color monitor, Pioneer 8210 video disk player, and our Micro–Ed controller box to link the micro and the video disk player. The controller box will soon be modified to permit its use with other micros and video disk players.

A key goal has been the development of a low–cost product for the mass market. Interactive video disk/microcomputer systems are not new. However, the price has been high. One highly touted configuration carries a purchase tag of more than ten thousand dollars. This may not be a prohibitive consideration for major corporations using large–scale industrial training programs. But that kind of money certainly prevents the employment of such a system by schools and families.

This is where LaserSoft comes in. The Micro–Ed controller box which is at the heart of the system will have a retail selling price of under two hundred dollars. Small in size (only slightly larger than a deck of playing cards), it enables the microcomputer to access at random any of the thousands of frames contained on the video disk and present these on a single color monitor screen utilized by both the micro and the disk player.

For example, Micro–Ed and the Society for Visual Education (a Chicago–based educational publisher of audio–visual materials) recently cooperated on an experimental project with far–reaching implications. More than thirty–seven thousand photographs from

various filmstrip series marketed by SVE were pressed on a single side of one video disk. Micro-Ed then wrote some educational programs that called upon a variety of these pictures for instructional purposes.

For instance, two of the programs had to do with understanding maps and globes. At one point, instead of laboriously trying to make the computer draw a picture of our planet as seen from outer space, we had the video disk present a photograph of Earth as it was actually seen on a moon voyage by American astronauts. It was possible to show pictures of a globe being taken apart to make a flat map. Highly detailed specialized maps of one kind or another were presented for examination. And so on. The difference in visual impact between using the microcomputer alone and employing it in conjunction with video disk technology was impressive, to say the least.

Micro-Ed's LaserSoft System is scheduled for commercial introduction by February, 1984. Dealer inquiries are invited and should be directed to George Esbensen, National Sales Coordinator. The toll free number is 1-800-MICRO-ED.

MICRO-ED Inc.
PO. Box 444005
Eden Prairie, MN
55344 612 944 8750

IEEE-488 to RS-232 Interface

The CmC BUSSter GR is a microprocessor based interface designed to allow any computer with an IEEE-488 interface to communicate with any device that has a standard RS-232 interface. These devices can include printers, modems, terminals, mainframes, and other computers. It is very versatile and easy to use.

BUSSter GR software controlled features include: selectable baud rates (75 to 19.2K), 110 character input buffer (larger buffers optional) with X-ON X-OFF feature, parity and word length selection, readable data terminal ready input, and data set ready output. Special commands allow the user to read input buffer status. An 8 position dip switch selects device address (0-31), start up baud rate (300 or 1200), auto line feed and auto carriage return.

The BUSSter GR is easily programmed through Basic commands from the host (IEEE-488) computer. It comes assembled and tested with case, power supply and standard female GPIB and DB-25 connectors.

The BUSSter GR sells for $495.00 in standard version, and is available from stock. Contact:

Shirley Fletcher
Connecticut microComputer
36 Del Mar Drive
Brookfield, CT
06804 203 775 4595

Interpod For VIC 20/ C64

Interpod is a free standing multiple interface designed to allow the VIC 20 and Commodore 64 to communicate with all Commodore peripherals and standard RS232 devices. It has serial bus I/O sockets and may be place anywhere in the serial chain. The unit does not affect the computer in any way and no memory is used inside the computer.

If an addressed device is not present on the serial bus, Interpod will attempt to find it on the parallel bus. A reserved device address (normally 4) may be chosen for communication with the RS232 port. All 3 buses can be used simultaneously and the selection process is completely invisible to the computer. The RS232 is also completely controllable. Contact your dealer or:

Oxford Computer Systems Ltd.
Hensington Road
Woodstock, Oxford 0X7 1JR
England tel. (0993) 812700

Letters

So many letters have been pouring in that it has become impossible to answer them all individually. Besides that, most of the questions have a similar nature. The result? Our new Letters section.

By the time this issue is released, all of our back mail prior to March 1 should have arrived. If you are missing anything, please write Mandy Sedgwick about getting things fixed up.

Oops on us: How stupid of me! What? What am I talking about? Oh yes, how stupid of me to believe that I should be subscribing to (and thus supporting) a Canadian (a genuine Canadian) publication. I thought in return for my few bucks of support that I would receive a bi–monthly (that means one every two months, right?) magazine. But alas, as with many things Canadian I have received very little for my money and am having to spend an additional amount (in time as well as money – how dare you take me away from my computer!) to inquire as to where my Transactors are. I have not seen one since the July issue (which came out in late August!?! – methinks somebody had too many coffee breaks). That was numero four of six. I should have number 5 and 6 by now. At least I think bimonthly means six in a year. Come on guys – what happened to number 5 and number 6?.

Bruce Spafford, St. Thomas, Ontario

Well Bruce, your letter pretty well sums up most of the letters in our complaints department, but with a sense of humour that is most appreciated. Your letter was dated Dec. 16, about 6 days after we sent out Issue 05. I admit, about 4 months passed between the release of Issue 04 and 05, and perhaps we spent too much time on one project. But looking back, I wouldn't have changed a thing. I also have to admit that I made the reference issue for my own benefit. So far I've used mine enough times to put dog ears on every page.

Mapped out: I have recently acquired a SuperPET and wish to program in the 6809 Assembler language. However, I cannot seem to find a set of memory maps.

Alan Wunsche, Oshawa, Ontario

Unfortunately, there is no such animal that I know of. The SPET languages are all written on a large mainframe that generates the 6809 code for the various languages. Memory usage is left at the discretion of the mainframe and since all development is done on the mainframe, the people at Waterloo have no need for memory maps. We actually tried to get them for our reference issue with no luck.

Tape worms: I have hooked up two Commodore VIC Datasettes tape to tape in an attempt to copy programs, however, the copy is totally ignored by the VIC. Audio wise, both tapes sound identical. An oscilloscope check of both show identical pulse widths, amplitudes, spacing, and AC & DC levels. A frequency counter check on the sync tone is also the same. Any suggestions?

R.D. Anderson, James Park, New Brunswick

Sounds to me like a case of bad luck more than anything else. With that many identical characteristics, rejection would seem unlikely. Richvale Telecomm have been copying tapes for TPUG since they began offering them using a high speed tape–to–tape copier with excellent results. Per-

haps they might be able to offer some advice.

Re–Numbed: Having worked with the HP 9830, which features auto line numbering as well as re–numbering, I wondered if the C64 had these features. I have experimented but so far without success. I noticed mention of a Renumber command in an earlier Transactor. Does this command exist?

Frank Vanzeist, St. Marys, Ontario

The Renumber command is not included as part of the C64 or any other earlier CBM equipment. In lieu of this, several programming aid type programs have been written that include this and other commands such as Find & Change and multi–line Delete. "Power" from Pro–Line software (see ad) is one of the more popular packages and I believe they have a version for the 64. Power loads into RAM and thus reduces the space available for writing BASIC, but not by much. Also, in Volume 4, Issue 03 of The Transactor, C64 Tiny Aid was published – another aid program that contains only a subset of these extended commands.

Explanation? Rebuttal? Retraction? Distraction?

I feel I must comment on the picture that appeared on page 14 of the last Transactor. Readers may be puzzled by it, since I appear to be in contrast with my usual image of a restrained, studious person.

Without an explanation of how the picture came about, the impression might be created that I was trying to "catch the spotlight", as it were. Nothing could be further from the truth. Karl Hildon, the illustrious editor of The Transactor, can vouch for the following facts. . . after all, he's in the picture (literally).

It happened at the World of Commodore show. I was in a place called the "media centre", which was a curtained off area in the vicinity of the Commodore displays. At the time, I was leafing through notes for my next presentation.

Suddenly, I heard familiar voices floating in from the other side of the curtain. Some of them were long lost Commodore Canada people. . . not lost literally, of course, but who had left Commodore Canada and for that matter, had left the Toronto area. I knew that Paul Higginbottam (currently working in Dallas) would be coming, since he was a speaker; but until I heard his voice, I didn't know he had arrived. David Berezowski was a complete surprise: I hadn't known he would be at the show.

It was obvious from the things they were saying that they were lining up for a photograph, there on the other side of the curtain. They didn't know I was there; and I had just that moment discovered that they had arrived.

I really didn't want to walk out and break up the session with a social scene ("Hello, Paul, when did you get here?. . . Good to see you, David . . ."). I felt I could make my presence known in a less disruptive way. I leaped through the dividing curtain in front of the group, faced the camera, and exclaimed "Ta–da!".

I hadn't expected the camera to take the picture. I expected even less that the Transactor would print it. If you looked at it with no other explanation, you would think that five people were posing nicely for a photograph, and one was mugging at the camera. And I'm not like that at all.

I hope that this explains things. It was just an event in progress caught by the camera. You should understand that I'm still the same thoughtful, studious and reticent person I always was. Karl will vouch for that.

There's just one thing that bothers me about the photograph, though. Less than a second before, I had burst unexpectedly into the picture from behind an adjacent curtain. Here's what puzzles me: Why does nobody in the photograph look surprised?

Jim Butterfield, Toronto

Jims' account of the circumstances is a pretty accurate one. But when I got the disk from Jim with his latest articles on it, the above letter was title "protest" in the directory. Actually, I think the picture turned out fabulous. Otherwise we would have printed another shot that turned out somewhat less than spontaneous – my personal favourite kind of picture. We were going to print the other photo we snapped of Jim changing into his T–shirt, but Cosmopolitan bought the negatives off us – the $5 was a real shot in our budgetary arm.

As for the last question. . . is it possible there was nothing to be surprised about? Energy levels at computer shows run extremely high – a tense atmosphere comes with the territory. Jims' presence at any show is surpassed only by his talent for relieving some of that tension, and his method displayed in the picture had to be his most effective to date. In fact, I hope someday I'll be able to release Version 2.0 of the same photo come the next computer show. I hope Jims' "Ta Da" will be the que for that shot too.

Bits and Pieces

Computenmachinen Blitzensparken

Screen dazzlers do only one thing better than tormenting a cathode ray. They demonstrate the lightning speed of machine language. The next batch of loaders are the creations of Richard Evers. Rich says, "They're really not to difficult to write, it's just that they usually turn out differently than originally planned."

The Brain

When this one was finished, it left the screen in such a state of disorder it could only be called The Brain. The code is self modifying, a no no in more serious applications, but it makes fast, compact code. The Brain checks the STOP key – we felt a good brain should at least do that.

```
2000 rem the brain 80
2010 for j=634 to 693 : read x : pokej,x : next
2020 sys 634
2030 data 169, 128, 133,  88, 169,   0, 133,  87
2040 data 168, 177,  87, 133,  89, 230,  89, 165
2050 data  89, 145,  87, 200, 208, 243, 230,  88
2060 data 165,  88, 201, 136, 208, 235, 206, 149
2070 data   2, 238, 123,   2, 173, 123,   2, 201
2080 data 132, 208, 213, 169, 128, 141, 123,   2
2090 data 169, 136, 141, 149,   2, 165, 155, 201
2100 data 239, 208, 197,  96
```

Screen Marquis

Just like any other marquis, except this only does one screenfull and it's a lot shorter. It too checks the STOP key.

```
5000 rem screen marquis 80
5010 for j=634 to 688 : read x : pokej,x : next
5020 sys 634
5030 data 169, 128, 133,  88, 169,   0, 133,  87
5040 data 160,   2, 173, 207, 135,  72, 173,   1
5050 data 128,  72, 173,   0, 128, 141,   1, 128
5060 data 177,  87, 170, 104, 145,  87, 138,  72
5070 data 200, 208, 245, 230,  88, 165,  88, 201
5080 data 136, 208, 237, 104, 104, 141,   0, 128
5090 data 165, 155, 201, 239, 208, 202,  96
```

The Boxer

This one's in BASIC, but it uses the special window commands that are only in the 8032. It also makes a great screen "set-up" program for Screen Marquis 80 or The Brain 80. Type in The Boxer first:

```
10 b=160:c=79:e=23
15 for d=0 to 11 : a$=chr$(d+219)
16 poke224,0+d:poke225,24-d:poke226,0+d
   :poke213,79-d:print" S ";
20 for a=1 to b : print a$; : next
40 for a=1 to e
50 print" sqU "a$tab(c)a$; : next
60 b=b-4 : c=c-1 : e=e-2 : next d
70 sys 634  : rem goto 15
```

Then change line 70 from GOTO 15 to SYS 634 as shown. If you have already tried The Brain or Screen Marquis, line 70 will activate it once The Boxer is finished. Otherwise you'll have to RUN one of the previous loaders to avoid crashing.

Screen Marquis 40

. . .is the same as the 80 column version. The only changes are the last address of screen memory, and a short time delay loop was inserted at the end to slow it down a bit.

```
5000 rem screen marquis 40
5010 for j=634 to 698 : read x : poke j,x : next
5020 sys 634
5030 data 169, 128, 133,  88, 169,   0, 133,  87
5040 data 160,   2, 173, 231, 131,  72, 173,   1
5050 data 128,  72, 173,   0, 128, 141,   1, 128
5060 data 177,  87, 170, 104, 145,  87, 138,  72
5070 data 200, 208, 245, 230,  88, 165,  88, 201
5080 data 132, 208, 237, 104, 104, 141,   0, 128
5090 data 160, 240, 162,   0, 232, 208, 253, 200
5100 data 208, 248, 165, 155, 201, 239, 208, 192
5110 data  96
```

Commodore 64 and VIC 20 Versions

Screen Marquis for the C64 and VIC 20 is only a little more involved. The colour table must also be scrolled every time the screen is. Otherwise characters tend to disappear whenever they are moved to a location that initially contained a space. Early 64s won't have this problem because they have a different Kernal ROM. Marquis 64 was designed to work with either Kernal.

The number shown in bold in line 5130 is the delay counter value. The lower this number, the longer the delay.

```
5000 rem marquis 64
5010 for j=828 to 924 : read x : poke j,x : next
5020 sys 828
5030 data 169,   4, 133,  88, 169,   0, 133,  87
5040 data 169, 216, 133,  91, 169,   0, 133,  90
5050 data 160,  39, 177,  87, 133,  89, 177,  90
5060 data 133,  92, 160,   0, 177,  87, 170, 177
5070 data  90, 133,  93, 165,  89, 145,  87, 165
5080 data  92, 145,  90, 134,  89, 165,  93, 133
5090 data  92, 200, 192,  40, 208, 230,  24, 165
5100 data  90, 105,  40, 133,  90,  24, 165,  87
5110 data 105,  40, 133,  87, 144, 202, 230,  91
5120 data 230,  88, 165,  88, 201,   8, 208, 192
5130 data 160, 240, 162,   0, 232, 208, 253, 200
5140 data 208, 248, 165, 145, 201, 127, 208, 160
5150 data  96
```

Marquis 20 is for unexpanded VIC 20s. The numbers in bold in lines 5030, 5040 and 5120 are, respectively, the screen start address high byte, the colour table start address high byte, and the screen end address high byte. For VICs with memory expansion that changes these addresses, change these 3 numbers to 16, 148, and 18, respectively.

Once again, the number in bold in line 5140 is the delay counter value. Make this smaller for a slower scroll.

```
5000 rem marquis vic 20
5010 for j=828 to 924 : read x : poke j,x : next
5020 sys 828
5030 data 169,  30, 133,  88, 169,   0, 133,  87
5040 data 169, 150, 133,  91, 169,   0, 133,  90
5050 data 160,  21, 177,  87, 133,  89, 177,  90
5060 data 133,  92, 160,   0, 177,  87, 170, 177
5070 data  90, 133,  93, 165,  89, 145,  87, 165
5080 data  92, 145,  90, 134,  89, 165,  93, 133
5090 data  92, 200, 192,  22, 208, 230,  24, 165
5100 data  90, 105,  22, 133,  90,  24, 165,  87
5110 data 105,  22, 133,  87, 144, 202, 230,  91
5120 data 230,  88, 165,  88, 201,  32, 208, 192
5130 data 160, 224, 162,   0, 232, 208, 253, 200
5140 data 208, 248, 165, 145, 201, 254, 208, 160
5150 data  96
```

The Brain for the C64 works the same way as the 80 column version, but is subject to the same problem as Screen Marquis. Except this time it is corrected by adjusting the background colour as opposed to the foreground colour. If you remove line 2010 before running this program, you'll notice that the spaces on your C64 screen seem to be unaffected. That's because the foreground colour of a space is the same as the background colour. The POKE in line 2010 changes the background colour to give the characters something to show up against.

```
2000 rem the brain 64
2010 poke 53281, 493-peek(53281)
2020 for j=828 to 887 : read x : pokej, x : next
2030 sys 828
2040 data 169,   4, 133,  88, 169,   0, 133,  87
2050 data 168, 177,  87, 133,  89, 230,  89, 165
2060 data  89, 145,  87, 200, 208, 243, 230,  88
2070 data 165,  88, 201,   8, 208, 235, 206,  87
2080 data   3, 238,  61,   3, 173,  61,   3, 201
2090 data   6, 208, 213, 169,   4, 141,  61,   3
2100 data 169,   8, 141,  87,   3, 165, 145, 201
2110 data 127, 208, 197,  96
```

The Plunge

The Plunge also uses the window features of the 8032 so it won't work on other Commodores.

```
4000 rem the plunge - 1984 r.t.e. the transactor
4010 for j=634 to 702 : read x : pokej,x : next
4020 sys 634
4030 data 169,  19,  32, 210, 255,  32, 210, 255
4040 data 169, 128, 133,  88, 169,   0, 133,  87
4050 data 168, 177,  87, 170, 232, 138, 145,  87
4060 data 200, 208, 246, 230,  88, 165,  88, 201
4070 data 136, 208, 238, 230, 224, 198, 225, 198
4080 data 213, 230, 226, 169, 147,  32, 210, 255
4090 data 165, 224, 201,  13, 208, 210, 165, 155
4100 data 201, 239, 208, 196, 169,  19,  32, 210
4110 data 255,  32, 210, 255,  96
```

Sequins

Sequins is another demo that shows how the cathode ray can often not keep up with the incredible speed of machine language.

```
100 rem sequins 80/40
110 for j=634 to 662 : read x : pokej,x : next
120 sys 634
130 data 162,   0, 160,   0, 254,   0, 128, 238
140 data 127,   2, 222,   0, 130, 206, 133,   2
150 data 200, 208, 241, 232, 208, 236, 165, 155
160 data 201, 239, 208, 228,  96
```

```
1000 rem sequins 64 - 1984 r.t.e. the transactor
1010 poke 53281, 493-peek(53281)
1020 for j=828 to 856 : read x : pokej, x : next
1030 sys 828
1040 data 162,   0, 160,   0, 254,   0,   4, 238
1050 data  65,   3, 222,   0,   6, 206,  71,   3
1060 data 200, 208, 241, 232, 208, 236, 165, 145
1070 data 201, 127, 208, 228,  96
```

Curtains

"Curtains" demos how the 8032 (SuperPET, and B Series) video controller chip can be altered to blank the screen. Register 6 of the 6845 controls the number of display lines on the screen. Normally it contains the number 25, naturally.

The 6845 video chip is controlled by 2 registers at 59520 and 59521 (55296 & 55297 on B Series). First 59520 is POKEd with the register number for which you want access. Then 59521 is POKEd with the value to be sent there. Both registers are write only so reading them with a PEEK will give unreliable results.

The demo also shows how text can be written to the screen while it is blank. The chip is poked with values of 1 through 25, but there's no reason why you can't go directly from 25 to 1. To stop the program, hit the SHIFT key. If you hit STOP you may find yourself left with half a screen. POKE 59521,25 will get everything back to normal.

```
100 print" S ";
110 for j=1 to 24
120 for i=1 to 79
130 print" + "; : rem fill screen
140 next i : print
150 next j
160 print" s "
170 poke 59520,6  : rem select reg 6
180 for j=25 to 1 step-1
190 poke 59521, j : rem write to reg 6
200 next
210 print" print on screen while blank
220 for j=1 to 25
230 poke 59521, j : rem reg 6 still selected
240 next
250 if peek(152) then poke 59521,25 : end
260 goto 180
```

Graphic Print

This next routine is rather useless the way it stands, but the part that plots the bar chart is simple and fast. The variable HT (height) could be replaced by data READ from a DATA statement.

```
10 sc=4448 : ln=22  : rem vic 20
20 sc=8032 : ln=22  : rem vic 20 w/exp
30 sc=1824 : ln=40  : rem c64
40 sc=33408 : ln=40 : rem 40 column
50 sc=34048 : ln=80 : rem 80 column
60 input " S enter a word ";a$
70 rem poke 53281, 12 : rem for c64
80 for i=1 to len(a$)
90 ht=asc(mid$(a$,i,1)) : lt=sc+i-ln/2*(ht-64)
100 y=sc+i : poke y+ln, ht+64
110 for j=y to lt step -ln
120 if j=lt then poke j,123 : goto 140
130 poke j, 97
140 next j, i
```

Modulo Counter

Paul Obeda of London, Ontario, uses this handy little counter that will go from 1 to any value of your choosing, and then repeat, without any IF/THEN statements and independent of any FOR/NEXT loops.

```
c = -c * (c < max) + 1
```

The statement (C < MAX) will yield a result of –1 when true and 0 for false. For example, if MAX = 12 and C starts at zero, –C will be multiplied by –1 since 0 is definitely less than 12. 1 is added and C now equals 1. Then –1 is multiplied by –1, plus 1 equals 2. And so on until C equals 12. –12 * (12<12) equals 0, plus 1 and we're back to the start.

Of course MAX could be any value, but so can +1. This could be replaced by any expression your imagination can conjure.

Reverse RVS

Setting Reverse character print is as easy as printing a RVS field control character. But there's another way that is somewhat uncommon but can be handy in the right circumstances. Suppose you want every second character of a message to be in Reverse field. Can you imagine all those control characters? Try this:

```
10 a$ = " some string "
20 for j = 1 to len(a$)
30 c = 1–c : poke 199, c
40 print mid$(a$,j,1);
50 next
60 print " Q " : rem cursor up
70 goto 20
```

Line 30 POKEs 199 with alternating values of 1 and 0. (Use 199 for C64/VIC 20 and 159 for BASIC 2/4) This is the RVS field flag for the operating system. When you print a RVS field control character, the OS does virtually the same thing. Location 199/159 is checked by the PRINT routine as it outputs characters.

One Line PET Emulator

Need to RUN some PET software on your Commodore 64? Try these POKEs courtesy of Jim Butterfield. Most of what they do is set the screen to $8000 (32768) and the boundaries of BASIC text space from $0400 to $8000. This will handle most programs, even those that POKE to the screen. But programs with SYS calls to machine language routines that perhaps rely on the operating system will give you trouble no matter what adjustments are made. C64–Link users will have to use the Link Relocator first.

```
1 poke56576,5:poke53272,4:poke648,128:
poke1024,0:poke44,4:poke56,128:print " S " :new
```

On Error Goto

This tidy little error trapping routine is another Butterfield original. Jim POKEs the code in at $CF00 where it's out of the way, but it's totally relocatable so it could be set up anywhere. Line 50 adjusts the Error Message Link to point at $CF00 (207*256+0).

First, the code tests for an error. If there isn't one, it jumps to warm start or READY. If there is one, the error number is stored in $030D (781) which is also used as the X register save for SYS. (The error number is held in the X register – see page 22 of The Reference Issue for a complete list of error numbers). Then the number 1000 is placed at $14,15 and the routine jumps to the GOTO routine in ROM. Note: the stack pointer is reset to $FA so all RETURNs and FOR/NEXT loops will be popped off the stack.

Line "1000" is determined by the two numbers shown in bold on line 15 (1000 = 232 + 3*256). If you want to use a different range of lines for your error trap, just change these two numbers accordingly. For example, line 50000 would be 80 + 195*256).

```
10 data  16,   3,  76, 139, 227, 142,  13,   3
15 data 169, 232, 133,  20, 169,   3, 133,  21
20 data 162, 250, 154, 169, 167,  72, 169, 233
25 data  72,  76, 163, 168
30 for j = 52992 to 53019 : read x
40 poke j, x : next j
50 poke 768, 0 : poke 769, 207
100 rem test program
110 stop
1000 x = peek(781)
1010 if x = 2  then print " you already opened
     that file, numskull "
1020 if x = 20 then print " you can't divide by zero,
     calculus breath "
1030 if x = 11 then print " type it right this time, ninny "
1040 print  " something else went wrong, probably
     your fault "
1050 end
```

Once installed, try executing OPEN 4,4 twice, PRINT 1/0, and any old syntax error. With this routine one can write a more informative and user–friendly error status reporter. Some errors could even be fixed for the user followed by re–entry to the program.

But Seriously Folks. . .

Coming up with new discoveries on your computer might

be personally rewarding and intellectually stimulating, but there's no immediate recognition. Now it's time for the other extreme – a fanfare for everything you try. Thanks to our Rick Evers, everytime you hit return with this little routine installed, you'll get a drum roll and cymbal finale.

The routine is linked in by the Input Vector of the 8032 type machines. Line 110 re–points the vector at this code, and it in turn transfers execution to the input routine. We'll have more on vectors, links, and pointers in the next issue, "The Transition to Machine Code".

```
100 for j=634 to 686 : read x : poke j,x : next
110 poke 233, 122 : poke 234, 2
120 data   8,  72, 138,  72, 152,  72, 169,  16
130 data 141,  75, 232, 169,  55, 141,  74, 232
140 data 169,   0, 133,   0, 141,  72, 232, 160
150 data   0, 200, 192,  21, 208, 251, 230,   0
160 data 165,   0, 201,   0, 208, 238, 141,  75
170 data 232, 141,  74, 232, 104, 168, 104, 170
180 data 104,  40,  76,  29, 225
```

Zoundz

This next sound effect for the C64 is from 12 year old Howard Strasberg of Toronto, Ontario. Notice how little code is required to keep the SID making sounds once it's set up properly.

```
10 s=54272
20 for l=0 to 24 : poke s+l, 0 : next
30 poke s+3, 8
40 poke s+5, 128 : poke s+6, 8
50 poke s+14, 117
60 poke s+18, 16
70 poke s+24, 143
80 for fr=1 to 24000 step 100
90 gosub 150
100 next fr
110 for fr=24000 to 1 step -100
120 gosub 150
130 next fr
140 run
150 poke s+4, 65
160 for t=1 to 4
170 fq=fr+peek(s+27)/2
180 hf=int(fq/256) : lf=fq and 255
190 poke s, lf : poke s+1, hf
200 next t
210 poke s+4, 64
220 return
```

aMAZEing quickies

This next couple of short snorts come from Chris Zamara of Downsview, Ont.

The below will work on any commodore machine, but make sure you're in upper case/graphics mode. Try this tiny program:

```
10 print mid$(" /\ ",rnd(1)*2+1,1); : goto 10
20 rem a shifted " n " and a shifted " m "
```

Or for a different effect:

```
10 print mid$(" /\X ",rnd(1)*3+1,1); : goto 10
20 rem the X is a shifted " v "
```

Here's a neat variable one.

```
10 get a$ : v=val(a$) : if v then m=v*2
20 if rnd(1)<.5 then print left$(" ////////////////// "
   ,rnd(1)*m); : goto 10 : rem 18 shifted " n " 's
30 print left$(" \\\\\\\\\\\\\\\\\\ ",rnd(1)*m);:goto10
40 rem 18 shifted " m " 's
```

After running the above program, press one of the number keys 1 to 9. See the different patterns that the different keys give.

CBM 4032 V2.2 Disable

For those of you with 80 column machines, you may or may not know about Chuan Chee's CBM 4032 program. It converts the 8032 to behave just like a 4032 making life a little easier when you want to run a 40 column program. Version 2.2 is the latest. Early versions had minor bugs and loaded in about 10 seconds. V2.2 is clean as a whistle and boots up in a split second.

If this utility gets more than a little machine time on your computer, you need not reset the entire machine to disable it.

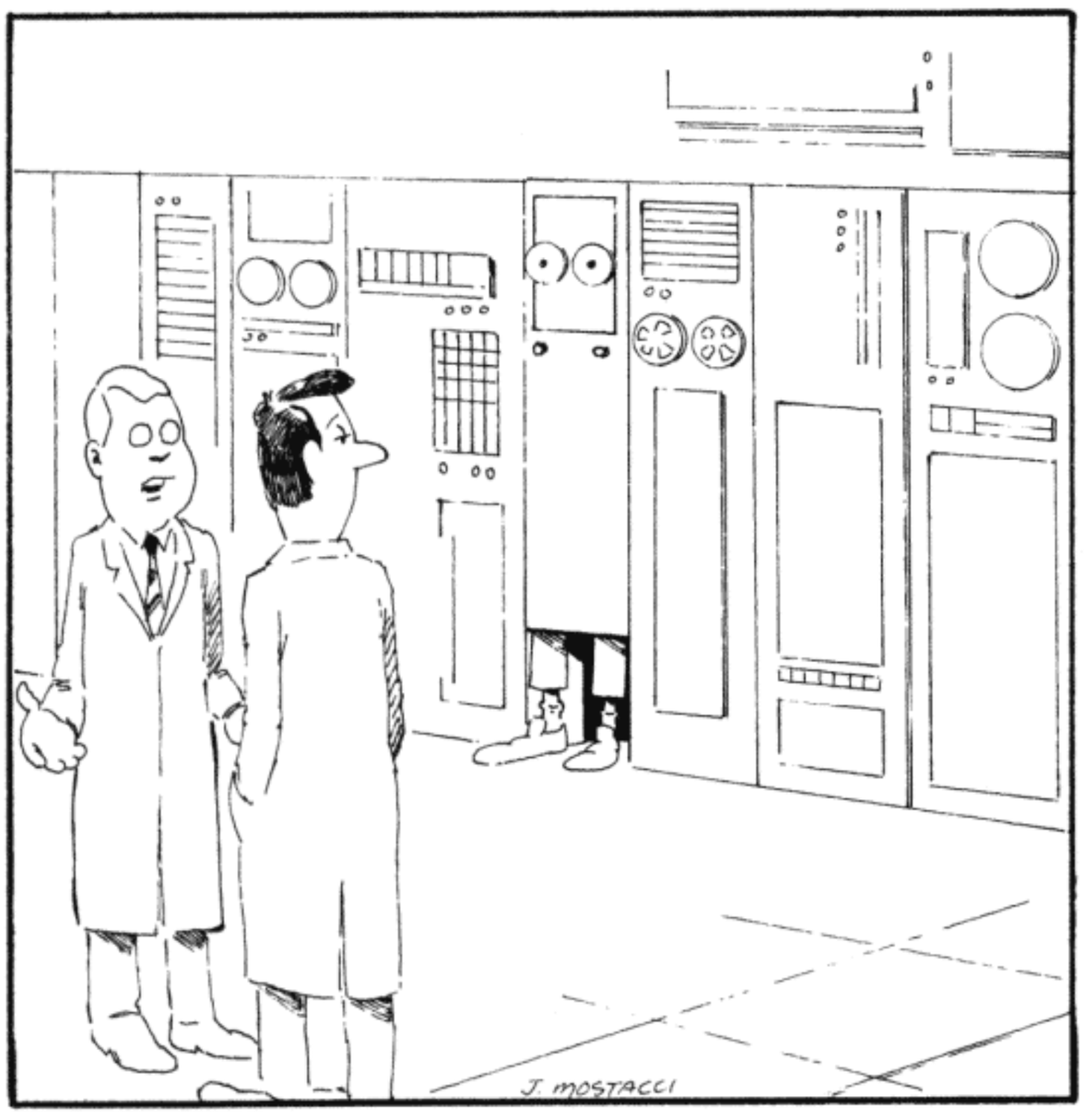
```
sys 14 * 4096
```

. . .disables CBM 4032 with no nasty side effects on the program in memory. Once back to 80 columns though, you'll have to re–boot to get back to 40.

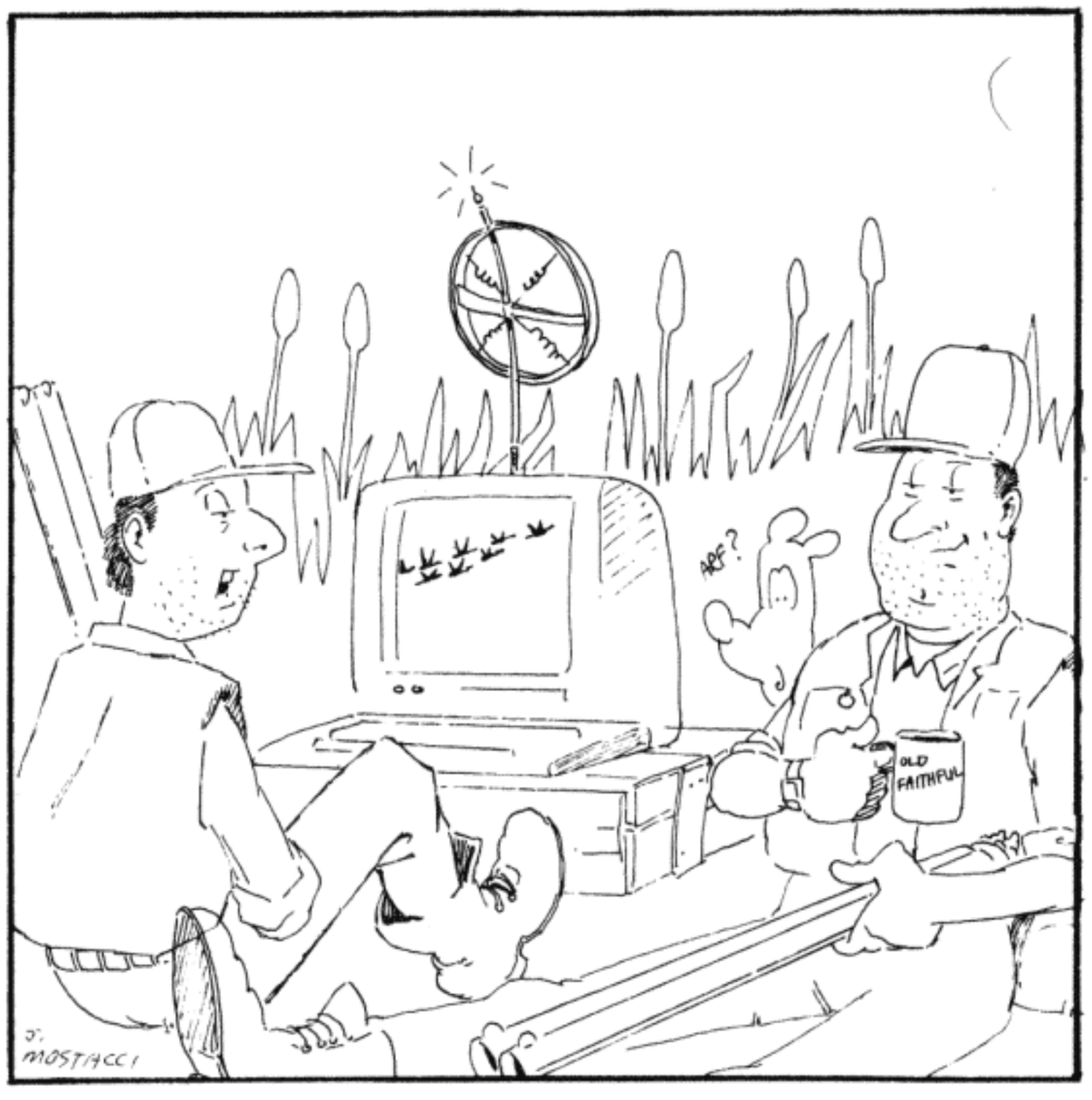
CompuKinks.

He may not look like much, but I'm tell'n ya Joe, he's an ornery cuss!

You say he's unbeatable eh? Here, slip this in his disk drive.



He'll do anything to get out of programming.



It won't be long now. There's a whole flock 20 miles SE of us heading our way

ARWUN

COMPUTER GENIUS

WEEK 1 – ARWUN COMMENCES WORK ON HIS SUBROUTINE TO SOLVE THIRD ORDER LINEAR EQUATIONS (NON-HOMOGENEOUS) WITH COMPLEX COEFFICIENTS AND REAL ROOTS FOR HIS COMPUTER MODEL OF THE GALAXY.

CLICKY CLICK CLAK CLAK CLICTY CLICK CLAK CLAK

WEEK 2: NEWTON AND EINSTEINIAN MECHANICS ROUTINE COMPLETED (ARWUN IS GETTING WEARY).

CLICKY CLICK CLACK CLACK CLICKY

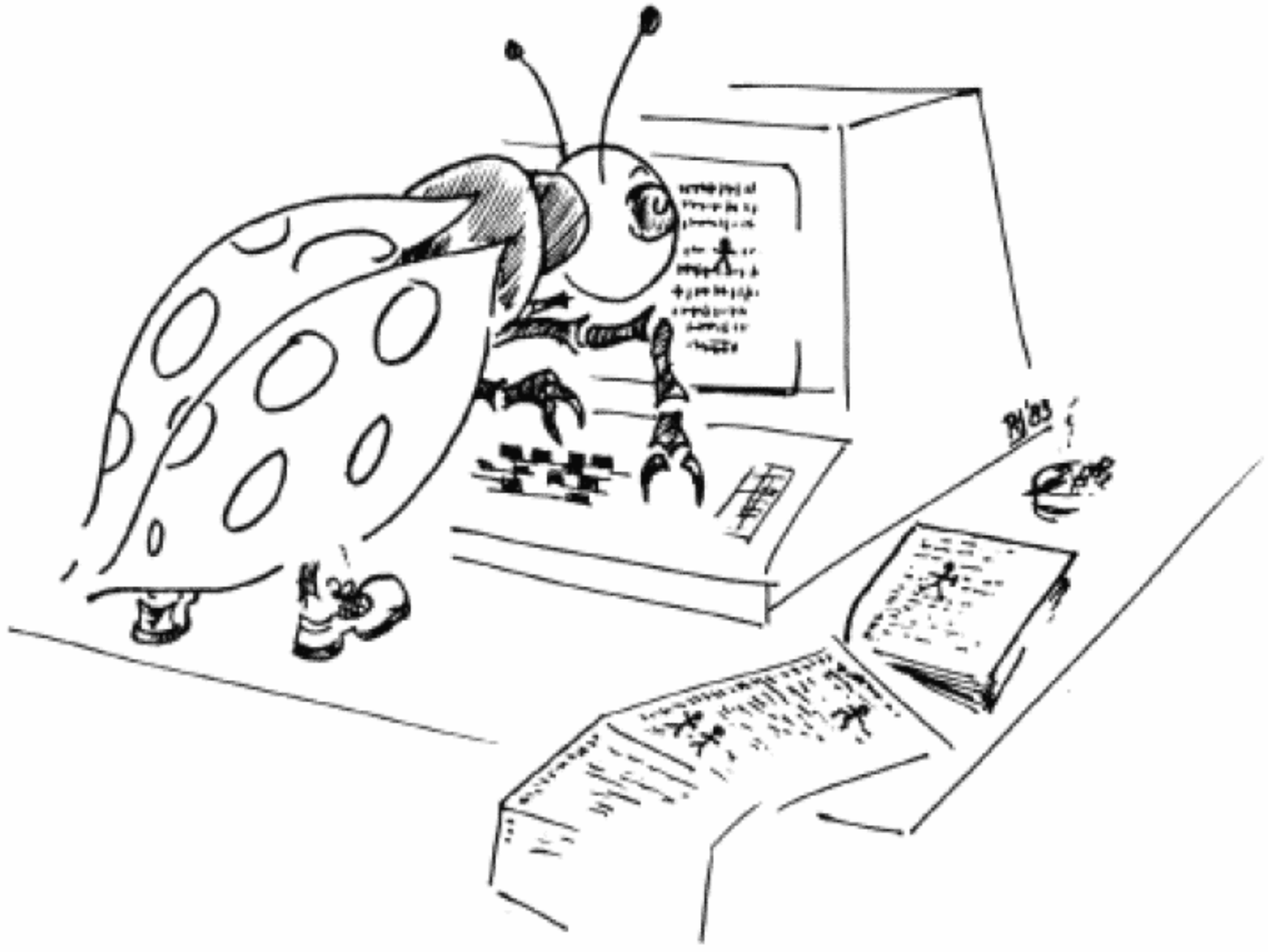
WEEK 3: ECSTATIC BUT TOTALLY EXHAUSTED, ARWUN HAS COMPLETED THE PRODIGIOUS TASK.

AAAAAAH!!

BUT THE NEXT DAY...

OH MY GOD! WHERE ARE THOSE DISCS?

THEY'RE ON THE FRIDGE WITH REST OF YOUR NOTES



© KURT LAUBER 1984

Permission Granted

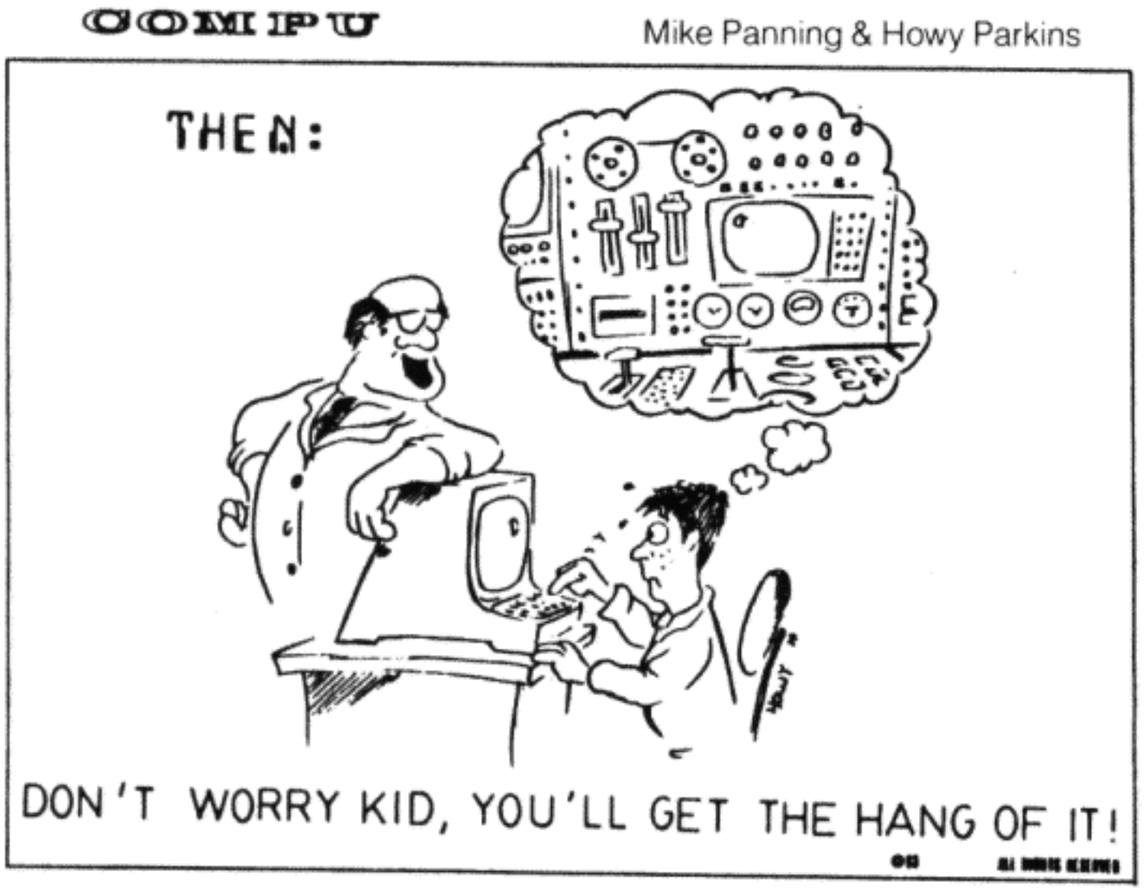
We get letters all the time requesting permission to reprint Fred Crone's Beagle Bros disk envelope warnings in various computer journals. Well, you've got our permission; thanks for asking! Here is a copy of the warnings, suitable for reproduction (unprotected!).

WE ONLY ASK that you print these drawings with both of the credit lines at the bottom. If you don't, and Fred finds out, he'll send his goons out after you.

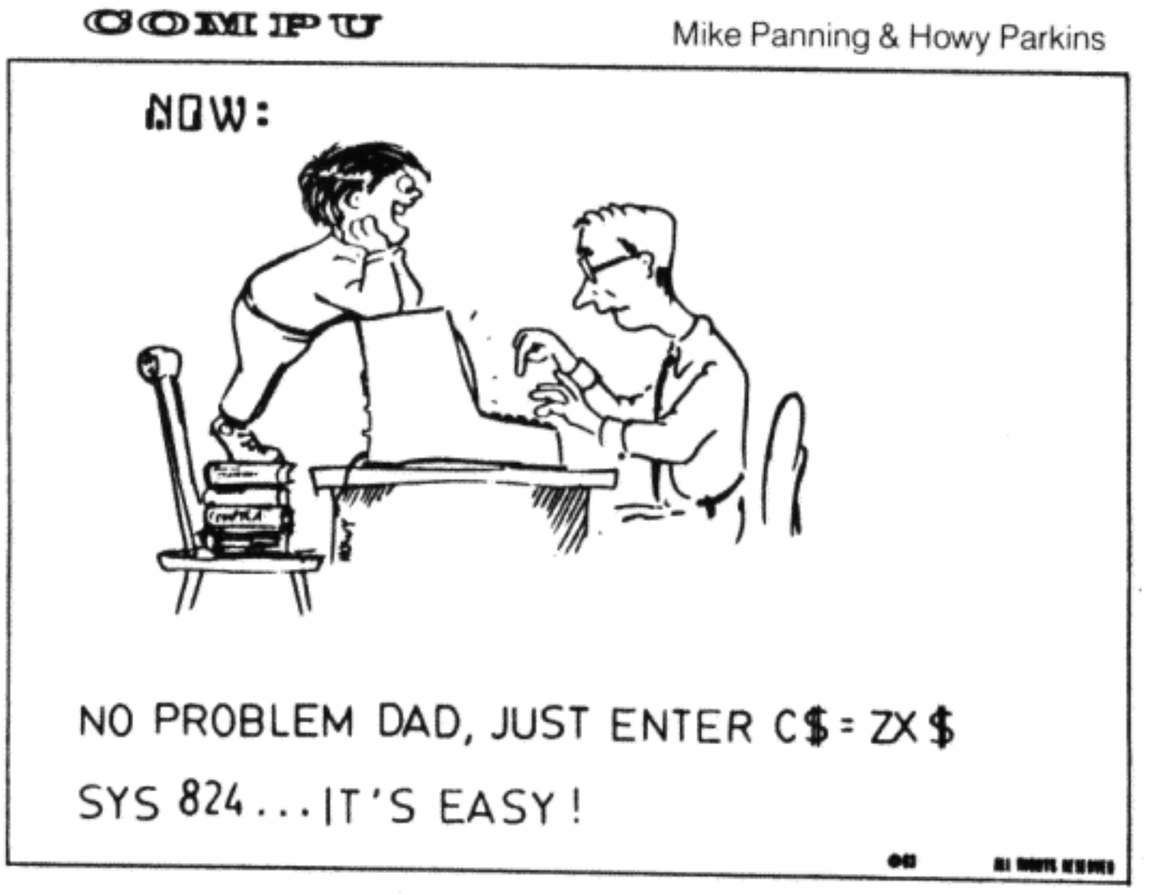
COPYRIGHT © 1983, BEAGLE BROS INC.
(BEAGLE BROS PUBLISHES APPLE II SOFTWARE.)

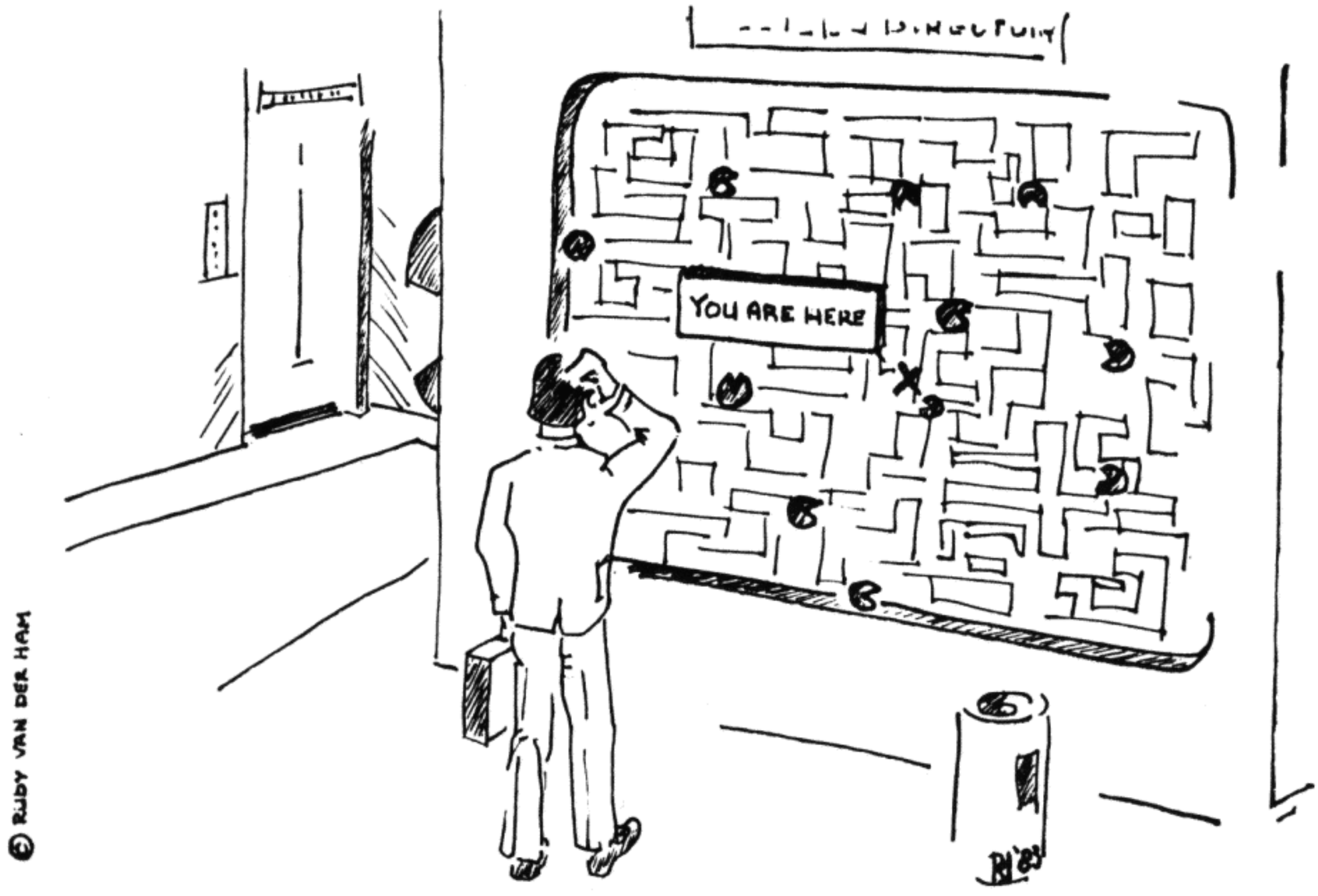
BEAGLE BROS / 4315 SIERRA VISTA / SAN DIEGO, CA 92103

COMPUTER — Mike Panning & Howy Parkins



COMPUTER — Mike Panning & Howy Parkins





The MANAGER Column

Don Bell
Milton, Ontario

Letters to THE MANAGER

There have been several letters to the MANAGER column. I will respond to a few of them here.

Alec E. Wittek, M.D. of the Department of Pediatrics, Stanford University Medical Centre asked about transferring data from one file to another using the 'Rearrange a File' function in the Manipulate Files option. This important process will be addressed in the next MANAGER column.

James K. Condon M.D. of Gloversville, New York keeps records of his cardiac pacemaker patients on a data base. He likes the 1500 character record capacity of the '64 MANAGER but had trouble with the REPORT GENERATE option. I hope that the column in this issue will resolve his difficulties.

Vernon H. Musselman of Royerford, Pennsylvania uses the '64 MANAGER for genealogy. He has over 2000 records on his family alone. He would like some help with the 'ARITHMETIC' option. This option will be covered in either the next or the following issue.

A '64 MANAGER Gift List – Part 3

Ever try managing fog, or just trying to drive in it? If you've ever used the manual for the '64 MANAGER you know what I'm talking about. It rivals those 1" pocket dictionaries as a miracle of brevity and conciseness. This is why I've gone to great lengths explaining this application in detail. Hang in there! A more comprehensive edition of the documentation is in the works and should be available by fall.

This is the 3rd and final part of the '64 MANAGER Gift List. But don't be deceived by the title. Although the application started out as a Christmas Gift List, it has evolved into a "Personnel File" which could easily be adapted to keep track of club memberships, customer or client lists. This application is useful for storing names, addresses, telephone numbers, important dates and Christmas list information. Information can be quickly and easily entered into the file, rapidly retrieved according to complex search criteria, sorted in alphabetical order, and many useful reports may be generated. It is probably helpful at this point to recap what we've covered thus far.

In the first article(Vol 4. Issue 05) I discussed loading the program, formatting diskette, designing an entry form on the screen in the CREATE/REVISE option, creating a file and entering records in the ENTER/EDIT option.

In the second article (Vol. 4 Issue 06) I listed the potential benefits to the user of this application, showed you how to get a screen printout of the lengths of fields in your file and field numbers, gave a detailed explanation of the fields in my revised screen format, showed some tricks for entering records quickly, and gave some examples of useful searches on information stored in the data base.

My original screen design needed to be slightly revised in order to produce some of the reports I wanted. Thus, I presented a revised screen format in the second article on the Gift List application. Unfortunately, revising your original screen may have inadvertently resulted in your losing all the records in the old file. I apologize for not explaining this in more detail in the last column and regret any inconvenience this may have caused you.

In this third article I will introduce you to the REPORT GENERATE option, showing you how to generate: (1) various mailing lists; (2) a backup copy of your address book; (3) party/gift lists; (4) selective telephone lists; (5) birthday and

anniversary reminder lists; (6) appointment or next contact date lists; (7) and finally, how to modify your screen and file to create a personnel, customer or client data base.

Error Correction

I want to get this off my chest right away. I would like to apologize for some typographical errors in the previous MANAGER column. The title of the column should have read "THE MANAGER COLUMN" not "THE COLUMN". Field numbers in the screen printout ran off the page and some extraneous blank lines appeared in "An Explanation of Fields In My Revised Screen Format".

To correct these errors, I will review the screen layout, field numbers, field lengths and some of the field descriptions. This will also be useful for those people who do not have the previous 2 MANAGER columns to refer to.

The following illustration shows the Enter/Edit form that was designed in the CREATE/REVISE option. Field prompts and field numbers are shown.

*** CHRISTMAS LIST (field numbers) ***

```
LAST NAME : ↑                    1                    ↑
MAIL NAME : ↑                    2                    ↑
ADDRESS   : ↑                    3                    ↑
CITY:       : ↑                  4                    ↑
STATE:      : ↑                5                ↑ M/F:6
COUNTRY   : ↑                7                ↑ S/M/D: 8
ZIP         : ↑   9   ↑          NEXT CONTACT: ↑ 10 ↑
HOME TEL  : ↑11↑  ↑  12  ↑   BUS:↑13↑  ↑  14  ↑
BIRTHDAY  : ↑15↑  ↑ 16 ↑ ANNIVERSARY: ↑17↑  ↑ 18 ↑
FAMILY      : ↑                  19                   ↑
NOTES       : ↑                  20                   ↑
INTEREST  : ↑                    21                   ↑

        **RECEIVED/GAVE LAST YEAR **
REC'D :      CARD: 22      PARTY: 23      VISIT/CALL: 24
GIFT  : ↑                25             ↑   $ ↑  26  ↑

GAVE  :      CARD: 27      PARTY: 28      VISIT/CALL: 29
GIFT  : ↑                30             ↑   $ ↑  31  ↑

              **GIVING THIS YEAR**
CARD: 32       PARTY INVIT: 33       VISIT/CALL: 34
GIFT  : ↑                35             ↑   $ ↑  36  ↑
```

The illustration below shows the lengths of each field, and the field type (A = alphanumeric, N = numeric).

*** CHRISTMAS LIST (field lengths & type) ***

```
LAST NAME : ↑                  30A                  ↑
MAIL NAME : ↑                  30A                  ↑
ADDRESS   : ↑                  30A                  ↑
CITY:       : ↑                30A                  ↑
STATE:      : ↑              21A              ↑ M/F:1A
COUNTRY   : ↑              21A              ↑ S/M/D: 1A
ZIP         : ↑ 8(A/N) ↑          NEXT CONTACT: ↑ 6N ↑
HOME TEL  : ↑3N↑  ↑  8A  ↑   BUS:↑3N↑  ↑  8A  ↑
BIRTHDAY  : ↑2N↑  ↑ 4N ↑ ANNIVERSARY: ↑2N↑  ↑ 4N ↑
FAMILY      : ↑                30A                  ↑
NOTES       : ↑                30A                  ↑
INTEREST  : ↑                30A                  ↑

        **RECEIVED/GAVE LAST YEAR **
REC'D :      CARD: 1A      PARTY: 1A      VISIT/CALL: 1A
GIFT  : ↑              23A              ↑   $ ↑  8N  ↑

GAVE  :      CARD: 1A      PARTY: 1A      VISIT/CALL: 1A
GIFT  : ↑              23A              ↑   $ ↑  8N  ↑

              **GIVING THIS YEAR**
CARD: 1A       PARTY INVIT: 1A       VISIT/CALL: 1A
GIFT  : ↑              23A              ↑   $ ↑  8N  ↑
```

Making Field Entries in the Enter/Edit Option

In the previous article I described how you enter information in the various fields in our ENTER/EDIT screen. Because of some typesetting errors in the text, I am going to repeat some of the field descriptions. Refer to these field descriptions when entering records in the ENTER/EDIT option.

Field 6(M/F) – enter 'M' for male, 'F' for female. Field 8(S/M/D) – enter 'S' for single, 'M' for married, 'D' for divorced. Field 9(ZIP) – this is a numeric field if you only use American postal codes, otherwise it is alphanumeric. Field 10(NEXT CONTACT) – enter year month day in a 6 digit format, e.g. May 8, 1984 is 840508.

Field 11(HOME TEL) – enter a 3 digit area code. Enter the normal telephone number using a space as a separator, e.g.

446 0001. Field 13(BUS) – enter a 3 digit area code for the person's business phone number. Field 14 – enter the normal business phone number using a space as a separator.

Field 15(BIRTHDAY) – enter the last 2 digits of the person's birth year, e.g. 46 for 1946. Field 16 – enter birthday month day in 4 digit format, e.g. 0508 for May 8. Field 17(ANNIVERSARY) – enter the last 2 digits of the year of the marriage. Field 18 – enter the month day of the anniversary in 4 digit format.

Field 19(FAMILY) – enter names of family members with ages, e.g. Johnny(4) Bill(7). . . Field 20(NOTES) – enter notes about family, or personal information.

Field 21(INTEREST) – enter your affiliation with this person, e.g. 'family' for family, 'business' for business associate, 'golf' for golf partner or 'club' member. It is important to use key words that describe a group of people you may want to put on a list, such as a telephone, card, letter, or party list. You can use as many words as you can fit in the 30 character field, separating words with a space.

The entries for fields 22 to 36 are fairly straightforward. For cards, parties, and party invitations, use 'Y' to indicate 'yes' and a blank or 'N' to indicate 'no'. For VISIT/CALL use 'V' for visit, 'C' for call and space for nothing. Gift descriptions and their approximate value will be helpful for budgeting and planning next year's gift list.

To obtain a screen printout of field lengths and field numbers for your screen. Select the Enter/Edit option from the main menu, then make sure the screen is clear (if not, press Shift CLR/HOME). To display field lengths, press 'up arrow'. To display field numbers, press Shift 'up arrow'. Press 'p' to get a printout either screen. You will need these printouts of field lengths and field numbers to refer to when doing searches or designing reports.

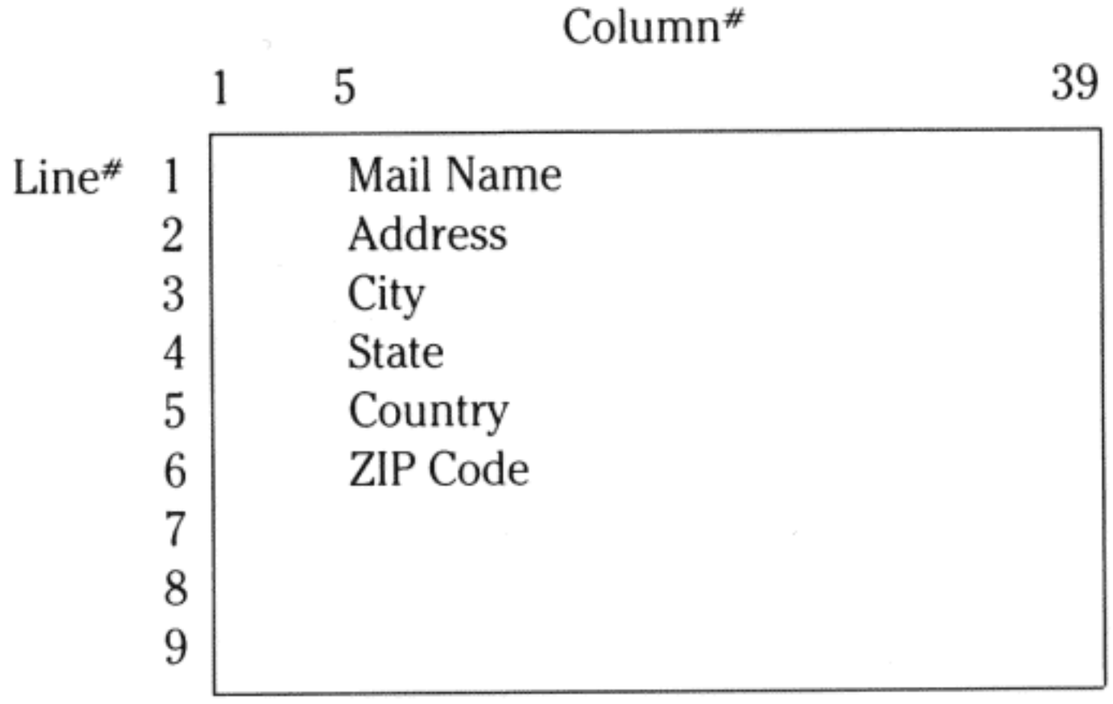
Planning Your Report On Paper First

We will use the REPORT GENERATE option to produce 3 kinds of reports:(1) name/address lists – for a mailing list, an address book, or selective people lists; (2) telephone lists in alphabetical order; and important date lists – for birthdays, anniversaries or next contact date.

It is a good idea to use the following 3 step procedure before actually entering screen information in the REPORT GENERATE option: (1) Sketch the report layout on paper; (2) divide the report into zones and print areas; (3) define the details of each print area.

1. SKETCH THE REPORT LAYOUT ON PAPER.
Draw your report using pencil and paper (preferably graph paper). Show titles, headings, 1 or more sample lines of the body of the report, and any footers or totals you want at the bottom of each page. Record on your drawing line numbers and important column numbers. For our name/address lists, I designed a report layout that would easily work on the screen and the printer.

REPORT LAYOUT



This first report is designed for mailing labels. Line#'s 7–11 are blank in order to advance the next mailing label on the printer to the correct starting position. If you print the list on 8" by 11" paper, then you will have 6 names on a page as 66 lines divided by 11 equals 6.

2. DIVIDE THE REPORT INTO ZONES AND PRINT AREAS
'64 MANAGER reports are described as having 3 different zones, a "HEADER ZONE" or title at the top of the page, a "LIST ZONE" or list of the contents of the report, and a "FOOTER ZONE" showing titles or totals at the bottom of the page. In a standard report we would have information for the Header, List and Footer Zones. HEADER information are page titles or headings that appear at the top of every page. LIST information is the main body of the report. FOOTER information appears at the bottom of every page of the report.

The name/address reports do not have a footer or header, only a list zone. The list zone consists of the name and address information drawn from records in our file.

Within each zone, there are several print areas ,i.e. distinct areas of information in the report. Print Areas can be any length up to 79 characters. One line may contain 1 print area or several. Everytime information in a report comes from a new source, it must be defined as a separate print area. The following illustration describes the zones and print areas used in our mailing list report. There are no print

areas used in the header and footer zones and 6 print areas used in the list zone.

ZONE AND PRINT AREA DIVISIONS

HEADER ZONE 0 Print Areas Used

Line#			
	1	Mail Name	(Print Area #1)
	2	Address	(Print Area #2)
	3	City	(Print Area #3)
	4	State	(Print Area #4)
	5	Country	(Print Area #5)
	6	ZIP Code	(Print Area #6)
	7		
	8		
	9		

FOOTER ZONE 0 Print Areas Used

3. DEFINE THE DETAILS OF EACH PRINT AREA
When you run the REPORT GENERATE option, you will have to define the details of each print area in each zone, by filling in a screen similar to the following:

REPORT GENERATE

LIST ZONE

PRINT AREA# AREAS OPEN

DATA TYPE (F–R–D–T): SUBSCRIPT:

TEXT/TITLE
. . . . . 1 . . . . . 2 . . . . . 3 4
LENGTH OF AREA: LINE NUMBER :

COLUMN NUMBER : CENTERING (Y/N)?

OF DECIMALS : ACCUMULATE (Y/N)?

BREAK TYPE (NONE, LNFDS, PAGE)?

In the screen above there are up to 10 pieces of information that you may need to supply. It's not as bad as it sounds, since you can often use the default entries for items (7) – (10).

(1) DATA TYPE (F–R–D–T) – the **source** of information for this print zone.

F = FIELD – information obtained from fields in your records
R = REGISTER – information obtained from arithmetic registers
D = DISPLAY – data specified by Arithmetic function
T = TEXT – text entered to describe the report or data

All of our information for this report comes from fields storing names and addresses of people. So all our entries will be 'F'

(2) SUBSCRIPT or source of the data;
If we selected a field or register data type, we will now supply the field# or register#. The subscripts for my report are 2,3,4,5,7,9, the field numbers for the name and address information I want on the mailing labels.

(3) TEXT/TITLE
. . . . . 1 . . . . . 2 . . . . . 3 4
If you specified text for DATA TYPE, then this is where you insert the text. The dots and numbers help you with the design of your text entry, i.e. 1 is at column 10, 2 is at column 20, etc.

(4) the LENGTH OF (print) AREA;
To keep this report simple, we have made all print areas 39 columns or 1 screen line.

(5) the zone LINE NUMBER;
Each zone has its own line numbers starting at line number 1. Print area line numbers refer to line numbers in a zone rather than line numbers on a whole page of a report.

(6) the starting COLUMN NUMBER;
Here you specify what column number your print area starts at. For my mailing list report, I started all print areas at column 5.

(7) CENTERING (Y/N)?
Whether the print area is to be centered;
Usually only header and footer print areas are centered.

(8) # OF DECIMALS
If required, this only applies to numeric fields.

(9) ACCUMULATE (Y/N)?
Whether you want to accumulate numeric information;

(10) BREAK TYPE (NONE, LNFDS, PAGE)?
If you chose to accumulate totals, whether you want to break the totals with 3 blank lines or a whole page.

If you are just learning how to use the REPORT GENERATE

option, you should always complete a CHART OF PRINT AREA DETAILS similar to the one below. The sources of information for this chart are all the previous diagrams in this article (i.e. field numbers, field lengths, the report layout and zone/print area divisions) . The chart below gives the details for print areas in the list zone of my name/address list. If your Create/Revise screen is different than mine, then your subscript numbers (field numbers) may be different.

CHART OF PRINT AREA DETAILS (LIST ZONE)

Print Area	:	#1	#2	#3	#4	#5	#6
Data Type (F–R–D–T)	:	F	F	F	F	F	F
Subscript (source field)	:	2	3	4	5	7	9
Text/Title	:						
Area Length	:	30	30	30	21	21	8
Line Number in Zone	:	1	2	3	4	5	6
Starting column#	:	5	5	5	5	5	5
Centering	:	N	N	N	N	N	N
# of decimals	:	0	0	0	0	0	0
Accumulate (y/n)	:	N	N	N	N	N	N
Break type	:	N	N	N	N	N	N

RUNNING THE REPORT GENERATE OPTION

Using the REPORT GENERATE option, you will be able to generate many different kinds of reports from this application. The reports generally fall into 3 categories: (1) name address lists; (2) telephone lists; (3) important date lists. I will take you through the name/address lists step by step and then give you pointers on how you can create the remaining reports on your own.

"Be of great patience, watch the red light on the disk drive, and read the messages at the bottom of the screen", is my advice to those who are just beginning to use the REPORT GENERATE option. At certain times in the program you may think nothing is happening. Be patient, if the disk drive red light is on, something is definitely going on.

While running the REPORT GENERATE option you need to refer to the previous illustrations in this article, i.e. field numbers, field lengths, the Report Layout, Zone and Print Area Divisions and the Chart of Print Area Details.

From the Main Menu, press 'R' for REPORT GENERATE
REPORT FROM KEYBOARD OR DISKFILE? Enter 'K', press <RETURN>

(Once the file is created enter 'D' instead of 'K', above. You can then recall the report from the disk using its Report file name.)

ENTER FILENAME? XMASLIST press <RETURN>
(This is the name of the file you created in Enter/Edit.)

ENTER SEARCH CRITERIA
This is the critical entry that decides who will be on any of your lists. The first time through enter no search criteria, i.e. just press <RETURN>. This will give you a mailing list of everyone in your file.

Referring to my screen layout of field numbers, here are some alternative entries, you may want to try later.

F3>' ' – only records with addresses
F32 = 'Y' – for this year's card list
F33 = 'Y' – for this year's party list
F34 = 'C' – this year's call list
F34 = 'V' – this year's visit list
F35>' ' – this year's gift list

IN ORDER BY INDEX, SORT OR FILE?
Enter 'S', press <RETURN>
(As the address book report requires alphabetical sorting, we will choose to sort on field 1, the last name field.)

ENTER NUMBER OF SORT KEYS?
Enter '1', press <RETURN>

Complete the next screen prompts as follows, pressing 'back arrow' after the last entry.

FIELD	LEN	ALPHA/NUM	ASCD/DESD
1	30	A	A

OUPUT TO SCREEN, PRINTER, DISK? S
Press <RETURN> to select output to screen
(Output the report to the screen until you get it the way you want it, then enter 'P' for printer instead)

ENTER LINE LENGTH 39
Press <RETURN>

ENTER NUMBER OF LINES PER PAGE 24
Press <RETURN>

(When you decide to output to mailing labels on the printer, enter '80' for 'LINE LENGTH' and '66' for 'NUMBER OF LINES PER PAGE'.)

Whether outputing to the screen or printer, make sure the 'NUMBER OF LINES PER PAGE' is correct or you may have problems.

You will now see the screen below:

REPORT GENERATE
DEFINE PRINT ZONES DEFINE HEADER – PRINT AREAS USED PRESS SPACE SELECTS/RETURN EXECUTES

At this point (although it is not immediately apparent), you actually have 4 choices, you can define the header, list or footer zones or you can exit to begin the report. Press the space bar several times and these choices will become apparent. For this report, we only have to define the LIST ZONE. Press the space bar until the screen displays DEFINE LIST, then press <RETURN>.

NUMBER OF LINES IN ZONE?
If you refer to the Zone and Print Area Divisions illustration, you will note that there are 9 lines in the List Zone.
Enter '9', press <RETURN>

PRINT ALL OR BREAK ONLY? A
Press <RETURN>

You will now be presented with the following screen:

REPORT GENERATE
LIST ZONE PRINT AREA# AREAS OPEN DATA TYPE (F–R–D–T): SUBSCRIPT: TEXT/TITLE . . . . . 1 . : . . . . . 2 . . . . . 3 . . . . . 4 LENGTH OF AREA: LINE NUMBER : COLUMN NUMBER : CENTERING (Y/N)? # OF DECIMALS : ACCUMULATE (Y/N)? BREAK TYPE (NONE, LNFDS, PAGE)?

For each print area in the LIST ZONE, complete the above screen using the information in my CHART OF PRINT AREA DETAILS. If the right entry is already displayed, you don't have to retype it, just press <RETURN>. When you have completed the screen for a Print Area press F7 to advance to the next Print Area. If you make a mistake you can void a Print Area by pressing Shift/CLR. When you have completed the screen for the last print area (in my case print area#6) press 'back arrow'. Note: You do not need to define print areas for the blank lines.

Your screen will now show:

REPORT GENERATE
DEFINE PRINT ZONES DEFINE LIST – 6 PRINT AREAS USED PRESS SPACE SELECTS/<RETURN> EXECUTES

Press the space bar twice or until the screen shows DEFINE EXIT then press <RETURN>.

ARE YOU SURE? Enter 'Y', press <RETURN>

SAVE THE REPORT CONDITIONS? Y
Press <RETURN>

IF YOU ARE DOING A SORT, ALWAYS SAVE THE REPORT CONDITIONS, TO AVOID POTENTIAL PROBLEMS.

If you are saving the report file conditions, you will be asked:

ENTER REPORT FILE NAME?
Enter 'NAME/ADDRESS' (or whatever you want to call the report)

SORT THE FILE? Y
Press <RETURN>
Follow the prompts at the bottom of the screen, swapping disks when necessary. The report will then print to the screen.

You do not have to re-enter all the report conditions the next

time you run the report as the report conditions were saved on a diskfile.

To return to the beginning of the REPORT GENERATE option press F2.

REPORT FROM KEYBOARD OR DISKFILE?
Enter 'D', press <RETURN>

ENTER REPORT FILENAME?
Enter 'NAME/ADDRESS', press <RETURN>

DO YOU WISH TO MODIFY THE REPORT?
If you enter 'N' the report will print to the screen according to the version you made the last time you saved the report conditions. If you want to change the report conditions specify 'Y'. You will now have a chance to modify any of the report conditions.

When you've got the report printing to the screen to your satisfaction, modify it so that it reports to the printer.

DO YOU WISH TO MODIFY THE REPORT?
Enter 'Y', press <RETURN>

Confirm all the previous report conditions by pressing <RETURN> until you reach the following screen prompt:

OUPUT TO SCREEN, PRINTER OR DISK
Enter 'P', press <RETURN>

Make sure the printer is connected, powered up and the paper or labels are positioned correctly. Thin mailing labels work better than thick ones. If the printer is not on, the program may crash.

Answer the next screen prompts as follows, pressing <RETURN> after each entry.

ENTER DEVICE NUMBER 4
ENTER PRINTER CONTROL CHARACTER 0
ENTER LINE LENGTH 80
ENTER NUMBER OF LINES PER PAGE 66

The DEFINE PRINT ZONES screen will now appear. Press space until DEFINE EXIT appears, then press <RETURN>.

Respond to the next screen prompts as you did before. The report should now print out. To abort the printout, press Shift/CLR.

You may have to fiddle with the total number of lines in the LIST ZONE to accommodate different sized printing labels, or printing paper.

Backup Copy of Name/Address Book

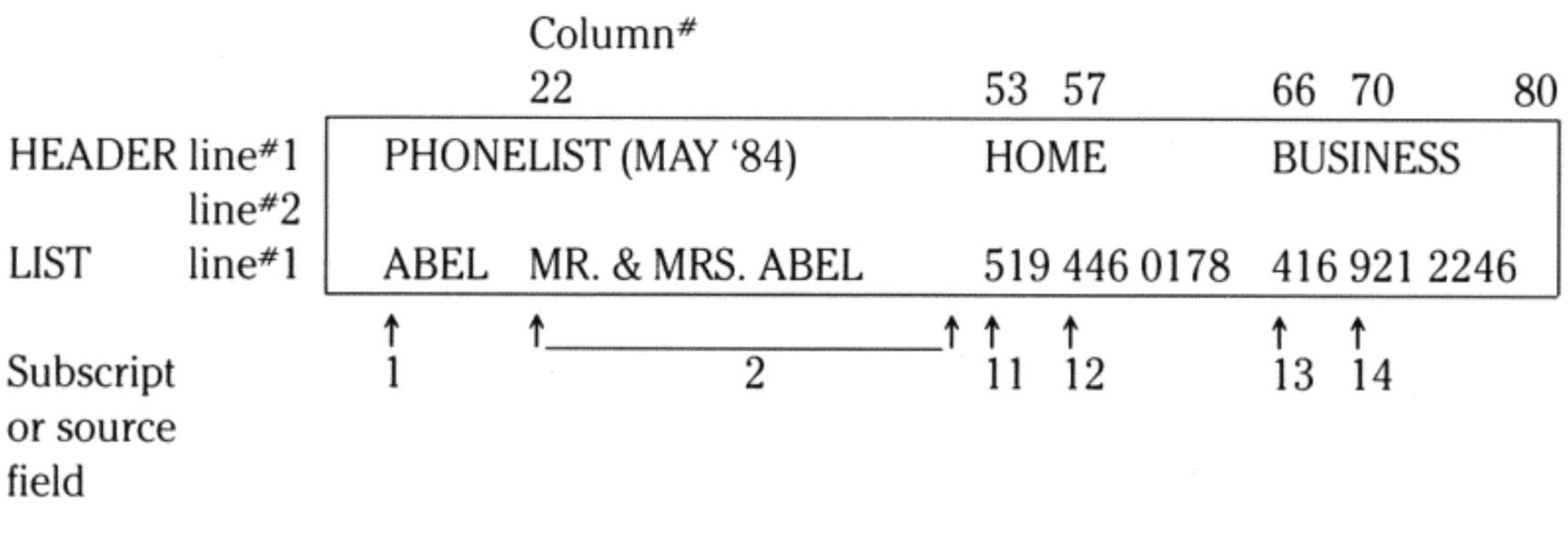
From the report conditions described above, you should be able to generate a backup copy of your address book, and mailing lists for cards, parties or visit/call lists. By changing the search criteria you can generate any number of other name/address reports

If you have entered all the names, addresses and phone numbers of your address book using the Enter/Edit option, you now have what amounts to an electronic backup copy of your name/address book.(If you want it, you can also make unlimited printouts of it.) The advantages of having your name/address book in a data base are: (1) entries are easily inserted, edited or deleted; (2) names can be sorted alphabetically anytime; (3) you can easily produce selective lists of names, addresses, phone numbers; (4) all relevant information about a person can be stored in one place; (5) you can produce lists or reports that relate to all the people in your data base file (e.g. all the single men or women you know between certain ages.)

Telephone Number Lists

Using the 3 step report design process mentioned earlier in the article (see Planning Your Report On Paper First), you should now be able to design a telephone list report for business and/or home phone numbers. By changing the search criteria you can generate telephone lists for different groups. For example, if all your friends who play golf have 'golf' in the 'interest' field in the record that describes them, then you could generate a phone list of all your golfing friends by using for your search criteria F21 = 'golf'. (or whatever field number you used for 'interest').

Here is one way of organizing a telephone list report.



Important Date Reminder Lists In Ascending Date Order

For birthdays and anniversaries, the month and day have been put in a separate field from the year in order to make it easy to sort and generate reminder lists (reports) by date. You can add a birthday date and/or an anniversary date print area to the mailing card list report, or preferably, design a whole new report. In your new report design the LIST ZONE might consist of the month/day field, the year field and then the mail name field. Your "SEARCH CRITERIA" will be "N#>0", where "N#" is the field number for the month/day for either the birthday or anniversary, and ">0" means you only want to search for those records that have an entry greater than 0. If you are using my screen format it will be "N#16>0" for a birthday list or "N18>0" for an anniversary list. To make the list even more valuable, specify the 'sort' option, sorting the month/day field(field 16 for birthdays, field 18 for anniversaries). Note: the report will not show a leading '0' in the date, e.g. '119' is really '0119' or January 19.

Using the "Next Contact" year–month–day field, you can generate a list of friends, family or clients you must contact in the near future. Numerically sorting on the "Next Contact" field in ascending order could produce a report showing the date, the person to contact, and any references you made in the "Notes" field.

In the "SEARCH CRITERIA", you can specify appointments on a specific date, between certain dates, before or after a certain date. If field #10 is the Next Contact date, then search criteria entries might be:

N10=840508 – May 8,1984 only
N10>840418 AND N10<840508 – between April 18 and May 8 1984.
N10>' ' AND N10<840508 – before May 8, 1984.
N10>840508 – after May 8, 1984.

Generating New Applications From Your Old File

You can use the CREATE function in the CREATE/REVISE option to revise the screen and file design to suit other applications, such as a customer or client list. You would probably want to replace all the Christmas list information at the bottom of the screen with more appropriate customer client information. You might even want to add another screen. If you want to use some of the information from your Christmas list file, you can transfer data from one file to another using the 'Rearrange' function in the MANIPULATE FILES option.

DON'T PHONE – WRITE!

If you have questions regarding this application or you would like to submit your own "terrific" application, please write me a legible, coherent letter, including your phone number, and a self–addressed return envelope with stamp. If you submit an application, send it on disk or at least send screen dumps of the ENTER/EDIT screen, a hand–drawn report chart and any math and sample data. I will attempt to answer letters either directly or in this column. Write to: Don Bell, c/o The Transactor, 500 Steeles Ave., Milton, Ontario, Canada, L9T 3P7.

P.S. I want to congratulate all readers who successfully developed this application. You deserve a "software perseverance" medal!

Four Wordprocessors For The Commodore 64

George Shirinian
Toronto, Ont.

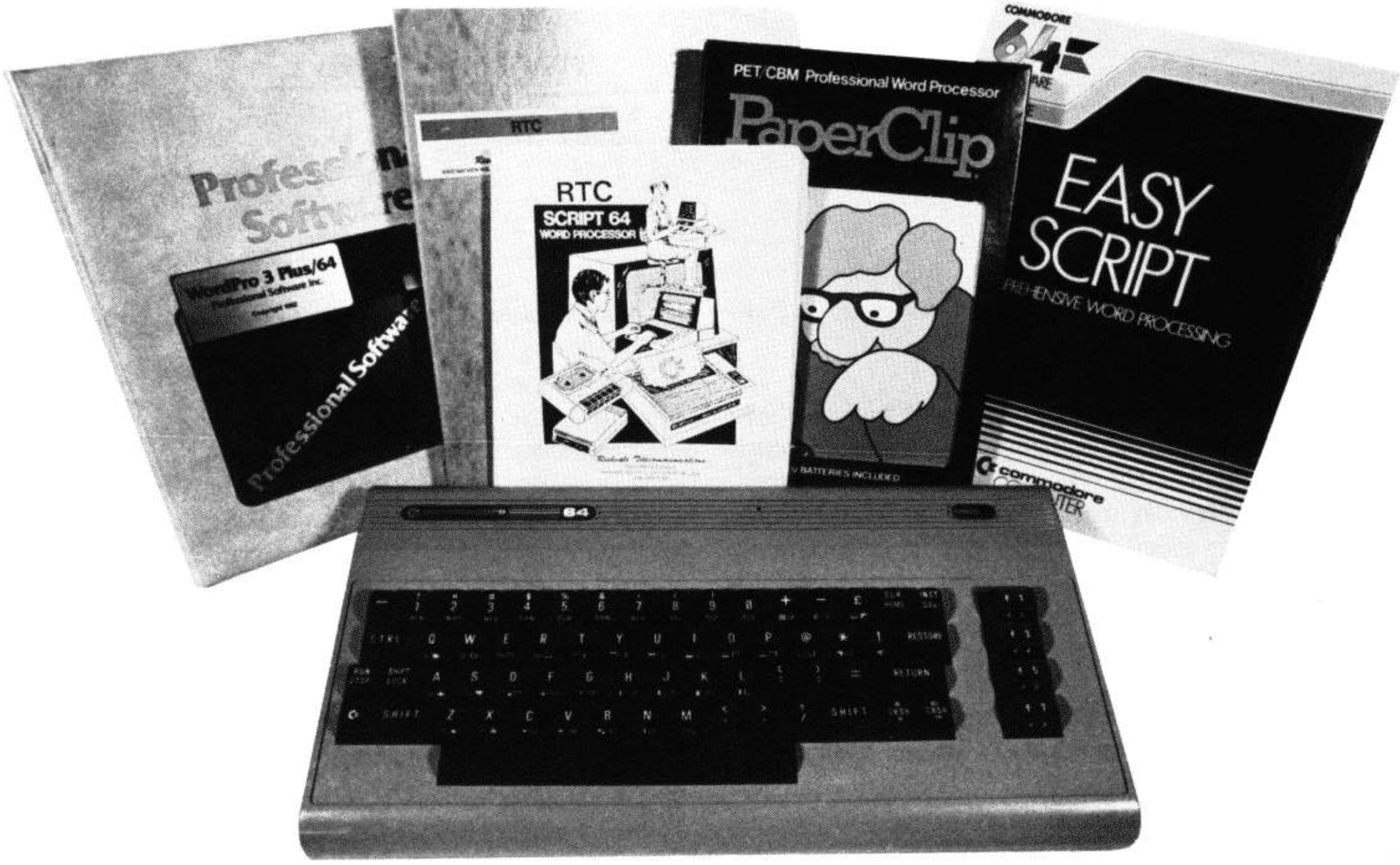


Photo by Roberto Portolese

Easy Script. Commodore Business Machines, 3370 Pharmacy Ave., Agincourt, Ont., Canada M1W 2K4. $99.95

PaperClip 64 (version C). Batteries Included, 186 Queen St., W., Toronto, Canada M5V 1Z1. $149.95

Script 64 (version 2.0). Richvale Telecommunications, 10610 Bayview Ave., Richmond Hill, Ont., Canada L4C 3N8. $129.95

WordPro 3 Plus/64. Professional Software Inc., 51 Freemont St., Needham, MA 02194, U.S.A. $69.95

Introduction

It is seems that among all the applications for personal computers, word processing has been and remains the most popular. While owners of the Commodore 64 have been spoiled by excellent sound capabilities and arcade–style graphics, it might seem unreasonable to expect such an excellent games machine to provide word processing functions worth writing about. There is the story of one wise guy who, when hearing about the wonders and benefits of word processing asked, "So what's your highest score?". As the user base of the 64 continues to grow, however, more and more people are interested in being able to use their machine for personal productivity.

Having worked closely with the four programs described below, I can happily report that a high level of sophistication in word processing exists for the 64. All word processors are designed to do essentially the same job. They facilitate the entering, deletion, rearrangement and printing of text. It is the special features which each program offers, therefore, that will receive particular attention in this review.

Documentation

A program's manual should be not only a reference to the program's functions, but, ideally, should also provide the user with a tutorial on how best to use the program. Any codes or conventions (i.e., ways of presenting information that are peculiar to the document) should be pointed out right at the beginning. A thorough index by subject should make it easy to double check anything of which the user is unsure. Physically, the manual should lay flat for easy consultation while at the keyboard, and not be so cumbersome as to occupy the whole desk. Finally, the manual should be well–written and easy to understand.

WordPro's manual comes in a well–produced, 3–ring binder. The 151 pages of instructions take a step–by–step tutorial approach and do not take for granted any previous computer or word processing experience. There were a few changes to the program since the manual that I received had been printed. These changes are listed in a 2–page photocopied insert, stuck in the inside front cover. Although the table of contents is fairly descriptive and there is a brief subject index, I found a number of functions that are not listed in either place (eg., assigning variable characters, duplicating lines). I also discovered functions the program is capable of, but that are not discussed at all, just listed in the summary (e.g., setting pitch, setting number of lines per inch). Otherwise, instructions are very clearly explained with good examples.

PaperClip's manual is a well–produced, spiral bound booklet of 174 pages. It lays open easily and is compact on a crowded desk. It is clearly written and is easy for a novice to use. A 13–page tutorial comprises the first chapter, while specific functions are detailed in chapters 2 through 15. There are 25 appendices, covering topics from creating a printer file to a summary of functions and commands. (Thank goodness in this second edition there is an appendix listing functions in alphabetical order!) A 7–page subject index completes the volume. Overall, the manual is thorough and rates an "excellent".

The author of the 139–page manual for Script 64 must be a good student of human nature, I thought to myself, as the first section is entitled "Load Script 64 and Start Typing". This section is for the impatient new user who wants to try the program out before studying the instructions, but it is then too bad that no mention is made until Section 2, that a "dongle" must be inserted before the program can be run. The rest of the manual is a careful, step–by–step guide to the program's functions. The references in the Table of Contents are to chapter and section, which are in the text, and not to page numbers, so it takes an extra bit of looking to find the information you want. There are 4 appendices for technical information about the program and a 7–page subject index, again by section number. Some numbers in the index do not match the numbers in text, e.g., for "Quitting". In any case, the subject index is a bit uneven and some topics are not detailed there. The manual is bound in such a way that it does not easily lay flat on its own.

fp

Easy Script's manual is easily the most attractive of the four. Physically, the book is coil–bound on the slightly larger "A" size paper standard in Britain (roughly 8 1/2 by 12 inches). The print is large, well–spaced and easy to read. Divided into two sections, the manual's first half is an introduction and tutorial, while the second comprises a reference guide. The 48–page tutorial section does not cover all the functions of which Easy Script is capable, but is clear, explicit and good for novices. A 5–page subject index makes the manual's contents readily accessible. A handy reference card listing functions and commands and which fits neatly into the disk jacket is provided. Also included were 2 separate pages of instructions, dated July 14/83, on using Easy Script with various printers and interfaces.

Copy Protection & Customer Protection

There are 3 basic methods of copy protection in the world of microcomputers. One is to require a chip or "dongle" to be inserted somewhere in the computer. The program checks for the presence of the "dongle" and refuses to continue if it is not present. Because the dongle is very difficult for the average person to copy, the software producer can afford to make the disk fully copiable. This is a great benefit to the software owner, as he can keep copies on a number of working disks and also be assured of having a backup copy safely put away.

A second method of protection is to change the order of the tracks on the disk. The disk is encoded with instructions telling the head where to find the information it needs on the disk, but this information is not accessible when using the Copy or Backup commands. This means that once your disk wears out or is damaged, you will have to pay for a new copy. Some companies will sell a backup for a fee, but this is still a great inconvenience.

The third method is my personal favourite: no protection at all! Unfortunately most companies have found it very uneconomical to rely on the honour system, so we are stuck with the ever–increasing prevalence of copy protection schemes.

WordPro and Easy Script use the disk–encoding method of

copy protection. Professional Software will provide a replacement copy for a damaged disk "within one business day at the appropriate exchange fee and return it to you via First Class Mail". Easy Script provides 2 copies of the disk in the package when you buy the program.

PaperClip and Script 64 use a "dongle" that plugs into joystick port #1. This allows backing up the disk. (By the way, do not be surprised if the Backup command does not work on your dual drive with the Paperclip disk. I had to use Copy to transfer the files onto an already formatted disk.)

Batteries Included and Richvale both continually revise and improve their programs. Both will update your disk at no charge if you bring it in and will sell you an updated manual for a nominal fee. While the method of providing for people who live outside of the Toronto area is mentioned by neither, I trust that some provision for exchanging disks by mail can be made.

The policy of continually upgrading programs and providing ongoing user support is to be strongly encouraged and companies that offer this should get full credit.

Size of Text

One of the big advantages of the 64 is its 39K of RAM available to the user. This allows reasonably large text files, as all of the programs here access data from RAM only. This is unlike other popular software, such as WordStar, which keep a file to disk open during editing, and therefore can have files as large as the disk's capacity. The disadvantages to keeping a file open to disk, however, are a great reduction in speed of access to the data, and great vulnerability to disaster should there be any interruption of power or the computer freezes up.

Using a simple formula based on average real–life use, a rough indication of the practical capacity of each program is offered. 1 screen line = 40 characters, 5 characters = 1 word, 40 characters = 8 words per line, 425 words = 1 printed page.

Easy Script allows 764 lines of text, which equals about 14 printed pages.

PaperClip allows 520 lines of text, about 10 printed pages.

Script 64 presents its text in "screens" of 22 lines each. Forty such screens are kept in memory at any one time. Forty screens contain 880 lines with the equivalent of roughly 16.5 printed pages.

WordPro contains a maximum of 329 lines of text in main memory, plus an additional 23 lines, minimum, in "extra text". The user is presented with the option of selecting the number of lines of text desired in main memory (minimum 179, maximum 329). Lines fewer than 329 are automatically assigned to a reserved area of memory called "extra text", which can be accessed independently of main text for such things as viewing a disk directory without erasing text, appending paragraphs, etc. This works out, then, to approximately 6 printed pages.

Note that the above figures are for 40–column video mode. Easy Script, PaperClip and Script 64 support video displays of greater than 40 columns, at the user's discretion. (More on this later.) In 80–column display mode, the number of lines of text is halved, but since the length of the lines is doubled, the amount of data remains the same.

Editing

All four programs use the "wordstream" method of entering text. This means, simply, that as you type out your words, they go onto the screen in one continuous stream. You do not end at the edge of your screen and hit return the way you would with a typewriter. The program assembles the words and makes sure each line ends with a whole word intact during output. You need press return only at the end of a paragraph.

Wordstream differs from the method of other popular programs, again, such as WordStar, which offer what is called "word wrap". Word wrap allows you to see on the screen exactly what you will see on paper. As you type, the program rearranges the words on screen before your eyes. This method is a little slower than wordstream, and I have heard the occasional person complain that it is distracting to have the text constantly moving around.

To be honest, however, it is very advantageous to know how your text will look before you go to the trouble of printing it out on paper. Three of the programs, Easy Script, PaperClip and Script 64, offer an output to video mode that allows just this. By outputting to video, you can see your text scroll by just as it would print out on paper. Since most writing is conventionally formatted to something wider than the 40–column display of the Commodore 64, these programs have made provision for horizontal scrolling in addition to the familiar vertical. Thus you can see your text, 40 columns at a time, up to 250 columns across, in the case of PaperClip, 240 in the case of Easy Script and 80 for Script 64. This facilitates the preparation of tables and charts especially.

In addition to the horizontal scrolling facility, two of the programs, PaperClip and Script 64 offer true 80–column display (at one time) in all modes. PaperClip has an 80–column version of the program on the disk that is designed to work with a hardware adaptor, such as the Data20 Video 80 Pak and Batteries Included's own Super–80 Pak. The Script 64 disk also includes a version that produces an 80–column display via software, with no add–ons required! While this is a little difficult to read, it does really work.

All four programs use the same style of status line display at the top of the screen in lieu of a more descriptive menu. Single letters on the status line are highlighted when you press the appropriate key or combination of keys to put you into one of the editing modes. This does require some effort on the part of the user to learn the numerous codes and key combinations. The status line also displays such information as line and column position of the cursor.

Table 1 lists most of the major editing functions available with a "y" under the name of those that offer them. Since most of the functions are common to all, let us look briefly at those which are less common.

Auto scroll, or "panning" as it is called in the Easy Script manual, allows the sybarites among us to view the text scrolling by without holding down any keys. Holding down the space bar in Easy Script increases scroll speed. Pressing any key in PaperClip halts the scroll, pressing any key again resumes it.

Bottom of text. It is often desirable to jump right to the end of your text, say when you have just loaded a file and want to pick up where you left off last time.

Breakpoint is unique to PaperClip. This function breaks up a word at the end of a line without inserting a hyphen.

Copy phrase. Easy Script and WordPro do not allow you to replicate less than a full line of text elsewhere in your document.

Defined keys. Being able to assign your own values to certain keys, often called "macros", is a great advantage. If, for example, there is a name or phrase that has to be repeated many times throughout your text, you can assign it to a couple of keystrokes, saving much time and effort.

Erase from cursor to bottom of text. It is worth mentioning that PaperClip erases not from the cursor position in this mode, but from the beginning of the line on which the cursor sits. Many times this is not what you would intend.

Erase from cursor to end of paragraph. This function, unique to Easy Script, does not close up the text when the lines are deleted. Instead, it leaves the space blank, presumably to be filled in with new text.

Erase screen. This function, unique to Script 64 simply clears text from the present screen of 22 lines. It is most appropriate to a screen–oriented program like Script 64.

Erase without deleting space. Part of "erase from cursor to end of paragraph", discussed above.

Go to specific line. Particularly useful with long text files where scrolling would be time consuming. This command could at least bring you near the spot you were interested in, even if you did not remember an exact line number.

Hyphenation. Script 64 will break a word longer than 6 characters in a relatively appropriate place when in automatic hyphenation mode, but is not always grammatically correct. The manual advises viewing the results during output to video first. If a hyphenation needs fixing, you can manually insert a "–" at which point the program will then hyphenate. The other programs leave it up to you to set where the break may occur and make their contribution by looking after whether a break is needed or not.

Linked phrase (hard space). Sometimes there are words that you do not want separated on two lines on the printout. This function ensures that the linked words remain on the same line.

Math functions. In addition to the traditional adding of columns, PaperClip also adds numbers in rows across the screen.

Move column. PaperClip has the unique ability to relocate an entire column to another location on the screen. It also allows you to erase a column without disturbing surrounding text.

Recover deleted text. Script 64, by virtue of its screen–oriented handling of text, is able to store in a buffer data that has been deleted. If you change your mind after deleting or rearranging some words , pressing the Shift and Back Arrow keys will restore the screen just as it was.

Set case of phrase. This useful function allows you to switch an existing portion of text from lower to upper case or reverse, without retyping it all.

Sorting. PaperClip allows you to sort columns into ascending or descending order. This can be done for a column

width of your choosing and without disturbing surrounding text. The speed of the sort slows down considerably with wide and long columns.

Spelling checker. Script 64 provides on the same diskette as the word processor, a spelling checker program. With this you can review your text for any misspellings or typographical errors before committing it to paper for a really perfect job! The program allows you to build a dictionary based on your own word usage. The more you use it, therefore, the larger it becomes. This dictionary must be maintained on a separate disk, unless you use the 8050 or 8250 drives. Spelling checker programs are great for the really miserable speller or typist, but do take some extra time to use and maintain. They are fairly popular, however, and sell separately for $50.00 or more. The inclusion of one with this program is a real bargain.

Table of contents. This is a rather neat feature for the book or manual writer. With the proper format command beside your chapter or section headings, plus the page number on which it occurs, PaperClip saves this information to a disk file when you do an output to video. The program is careful to gather this information only during the first pass on output. When the entire book is finished, you load in the file, format it and you have a table of contents all ready.

Word count. This feature, unique to PaperClip is a real blessing for writers. During output to video or printer the number of words in the file is displayed on the status line. Students who must write a 3000–word essay now can have a clearer idea of where they stand. Authors asked for an article "not to exceed 1500 words" can now stop fretting. And freelance writers who are paid by the word can now be sure of how big a cheque they can expect!

WordPro is the grandaddy of word processors for Commodore computers. Consequently its editing and formatting conventions have become more or less standard. Script 64 uses a unique approach to the handling of data. First of all, formatting is handled by control maps that set margins, page length, etc. These control maps are accessed in separate screens outside of the text.

As mentioned earlier, Script 64 accepts text in distinct screens. This has the advantage of allowing you to retain chunks of data separate from the rest of text and print them out in different documents or places. There is a significant disadvantage, however, in that you have to manually advance to the next screen by pressing the F1 function key every 22 lines. I found myself typing away and forgetting this limitation. Some of my words were lost before I looked up and realized it. Although the program does display the word "LIMIT" and provide an audible alarm when you reach the screen's limit, I am a touch–typist and am not used to not watching the screen at all times. I also prefer to keep the 64's sound off while word processing. Others may not be bothered by this.

Printing

Of course the goal of word processing is to get your words down on paper, so the program's ability to support your printer is very important. Many printers have very sophisticated capabilities that must be addressed by the software via control codes.

There are two different approaches taken by these four programs. WordPro and Easy Script address the printer from within the program. They are particularly designed for Commodore printers. ASCII printers are supported generally, but any special control codes must be specifically embedded in text by the user. Script 64 has a number of the most popular printers already configured in its program, but, like PaperClip, also uses separate printer files that are loaded into the program. This latter approach allows a high degree of customization for your particular printer. PaperClip offers dozens of printer files on its disk for many popular models, and, along with Script 64, provides special programs with which you can create your own file and merge it permanently with the word processor.

Table 2 lists most of the major printing functions these programs offer. Some of the more interesting ones are commented on below.

10, 12, 15–pitch. Script 64 will produce only 10–pitch on Epson printers. In 10–pitch it does a bold print that is certainly attractive. When you select 12 or 15–pitch, however, you are given normal 10–pitch.

Characters not on keyboard. Certain commonly used characters, such as the cent sign, do not exist on the 64's keyboard. These are accessible by assigning their ASCII value to a specified key. Each program handles this important function slightly differently, but they all work fine.

Continuous printing. Easy Script and PaperClip have an extra feature of changing to continuous printing in the middle of a run when you did not specify it at the beginning.

Device number. Easy Script allows you to have more than one printer connected. (Some interfaces allow this very conveniently). Since two devices can not have the same number, you can change one of them from within the

program.

Double spacing. While all of these programs provide a command for double and triple spacing, none offers 1.5 lines at a time without having to change the control code for number of lines per page. Although many daisy–wheel printers can be set manually to 1.5 spacing, most dot–matrix units require this to be set via software.

Double strike. By passing over the same line twice, a darker and bolder impression can be made. This is useful for headings or when emphasis is desired, or when a single pass just does not seem dark enough. (Maybe it's time to change your ribbon!)

Foreign characters. Many dot–matrix printers allow you to print non–English characters. PaperClip gives you the ability to create your own printer files in which you can define specific characters of any type. In order to see these characters on the screen, (e.g., the Greek alphabet), you must use a character set editor, available separately from other sources. As a convenience, a file to display the several distinctive French characters on screen is included on the disk. To invoke these characters you simply press the back arrow followed by one of the number keys.

Script 64 provides French characters as part of the main program, via the function and "@" keys. Printer support for French is offered from the printer menu, providing your printer is capable of generating the various accents, etc. Other alphabets are not supported.

Lines per page. PaperClip has a bug in that it does not count the number of lines per page correctly when a "header" command is used and the header is followed by a number of blank lines, e.g. hd4:TITLE. It goes ahead and prints the number of lines specified in the format line, e.g. pg55, but ignores the 4 blank lines that are specified after TITLE. Unless you adjust for this yourself by reducing pg to 51, your page will be too long.

Offset. This command allows you to shift the printing to the right if it does not fall just where you want on the paper. This saves you from having to reset the margins or physically shifting the paper. Easy Script's manual suggests using this facility for printing in columns by putting the same piece of paper through and printing out the text in several passes, shifted over each time.

Page length ignore. If you are printing a rough draft, it may not be important to you if the printing goes right over the perforation in your pin–feed paper. This function ignores the page length setting.

Paragraph indentation. PaperClip and Script 64 allow you to set the number of spaces the beginning of a paragraph will be indented during print–out, even if it was not indented during editing.

Paragraph spacing. Script 64 allows you to select the number of blank lines between paragraphs during printout, even if none were entered on the screen.

Semi–Proportional spacing. When right justifying text, most programs pad out each line with extra blank spaces between words. This can look sloppy under certain conditions because some words will have a normal single space between them while others will have several. Typographers have long had a scheme to measure the distance between letters and words in "micro–spaces". Although the space between letters is always the same, PaperClip is able to use these micro–spaces to pad out the blank space evenly between all words on a line that is right justified, giving a more professional and eye pleasing appearance. Few printers available for microcomputers offer semi–proportional printing, but PaperClip will support those that do.

Ragged left. You are given the option to justify the text on the right side only, and have the left side of the page uneven.

Reverse. Easy Script is capable of producing reverse field characters on Commodore dot–matrix printers.

Shadow. During a second pass over the same line, Easy Script will strike over the text but shifted slightly to the right, leaving a tiny bit of white showing between the two sets of characters, creating a "shadow" effect.

Underlining. It is worth noting that PaperClip has a special command to underline a blank space, as for creating a line for a signature, in addition to underlining characters. There can be some difficulty underlining a blank space in some of the other programs.

Take note that WordPro will work with printers attached only to the 64's serial bus. The other programs will support serial, parallel and IEEE printers which have been properly interfaced.

File Handling

Although it has become something of a cliche that to have a really serious computer system you must have disk storage, Easy Script, PaperClip and Script 64 allow the tape user to enjoy the benefits of word processing, as well as the disk user.

Of course all programs must be able to load and save files, but other disk commands are not always allowed, and when they are, it is not always in normal Commodore syntax. Table 3 lists the disk commands supported by each program. Below are some comments on a few of them.

Change device number. PaperClip allows you to set the disk device number if you are using a second drive, or if you have one with a device number other than 8.

You can not call up a directory of your disk in Script 64. In order to know the contents of the disk, the program requires an index file to be created by the user. Furthermore, the program stores its data in files automatically labelled "A", "B", etc., so, in the absence of recognizable file names, the user must also keep, manually, a separate list of what each screen contains.

When Easy Script and WordPro load a directory, it overwrites all other data in memory. If your directory is the right size, you can load it into WordPro's extra text area safely. Even though PaperClip loads a directory without disturbing the data in memory, the way the directory is loaded does not allow it to be printed out.

In order to minimize the possibility or read and write errors, WordPro allows you to initialize the disk drive at any time. This is especially valuable if you change disks during a session. For those using programs that do not offer this feature, I would sincerely recommend reading a disk's directory the first time you insert it, to give the drive a chance to initialize.

WordPro's conventions have more or less become the standard in many cases. One of the most visible is the way it saves data as program files. This is not the place to go into a technical discussion of file handling, just recognize, for now, that program files are different in nature from sequential files. Now, many other types of programs, e.g., mailing list handlers and database managers, save their data in sequential files. Sometimes you would like to take data from, say, your mailing list and incorporate into the text of a letter.

There are two ways of doing this. With Easy Script and PaperClip you could simply merge the desired sequential information directly into memory. They also allow you, while WordPro requires you, to read information from a sequential file, one field at a time and print it out as a variable block of information. In this case the two files are merged during printing, but not into one new file. Conversely, you can also save data from Easy Script, PaperClip and WordPro as a sequential file that can be loaded into another program or transmitted via a modem. Script 64 also provides the variable block function, and, in addition, has a separate utility program to convert an sequential file so that it can be loaded into memory.

When discussing the compatibility of files, it is worth mentioning here that many programs over the years have made efforts to be WordPro compatible. PaperClip uses WordPro's custom of saving program files, and reads WordPro files. It is incompatible with WordPro, however, in the following ways.

Certain embedded formatting commands can be linked on the same line, separated by a colon, in any order in WordPro. PaperClip requires some of these commands to be the first on the line or it does not recognize them, e.g. cn0 to turn centering off. It is a simple matter to go into a file and make the appropriate changes throughout, but an extra chore, nevertheless.

PaperClip saves some extra information in the way of screen tabs and other screen information that appear as miscellaneous characters when loaded into WordPro. There is a forced indentation of 21 blank spaces at the beginning of each paragraph that have to be manually deleted. Thus, while PaperClip is able to read WordPro files with a minimum of difficulty, it has been made very inconvenient for WordPro to read PaperClip files.

Other Features

PaperClip is programmed to work with The ARBITER and other multiple computer systems. WordPro provides a printer spooling feature that has particular application in a multi–user environment.

Easy Script has an additional utility program that relocates the 64–Link so that it does not interfere with the word processor. Full instructions are included.

In addition to the spelling checker program mentioned earlier, Script 64 also includes the RTC Scratchpad mail list and database management program.

Error trapping is an important consideration when judging any program. All four of these packages provide generally excellent error trapping. Unfortunately, Script 64 will lock up irretrievably if you attempt to output to a serial printer when one is not connected.

TABLE 1: Table of Editing Functions

	Easy Script	Paper Clip	Script 64	Word Pro
Adjust screen colour	y	y	y	y
Auto page numbering	y	y	y	y
Auto scroll	y	y		
Bottom of text	y	y		
Breakpoint		y		
Centering	y	y	y	y
Copy line	y	y	y	y
Copy phrase		y	y	
Defined keys		y	y	y
Delete line	y	y	y	y
Delete word or phrase	y	y	y	
Delete sentence	y	y	y	y
Enumeration			y	
Erase from cursor to bottom	y	y		y
Erase from cursor to end of paragraph	y			
Erase screen			y	
Erase without deleting space	y			
Fast scroll	y	y		y
Forced page	y	y	y	y
Global search	y	y	y	y
Global search & replace	y	y	y	y
Go to specific line	y			y
Hyphenation	y	y	y	y
Indent	y	y	y	y
Insert mode	y	y	y	y
Insert line	y	y	y	y
Linked phrase	y	y		y
Margin release	y	y	y	y
Math functions		y	y	y
Merge or append files	y	y	y	y
Move column		y		
Move line	y	y		y
Move phrase	y	y	y	
Move text to another file	y	y	y	y
Next screen	y		y	
Numeric tabs (decimal justific)	y	y	y	y
Previous screen	y		y	
Recover deleted text			y	
Search	y	y	y	y
Search & replace	y	y	y	y
Set case of phrase		y		
Shift lock	y	y	y	y
Sorting		y		
Sound	y		y	
Spelling checker			y	
Table of contents		y		
Tabs	y	y	y	y
Top of screen	y	y	y	y
Top of text	y	y		y
Word count		y		

TABLE 2: Table of Printing Functions

	Easy Script	Paper Clip	Script 64	Word Pro
10–pitch	y	y	y	y
12–pitch	y	y	y	y
15–pitch	y	y	y	y
ASCII printers	y	y	y	y
Bold	y	y	y	
Centering	y	y	y	y
Characters not on keyboard	y	y	y	y
Continuous printing	y	y	y	y
Device number	y			
Double spacing	y	y	y	y
Doublestrike		y	y	
Enhanced	y	y	y	y
Foreign characters		y	y	
Headers & footers	y	y	y	y
Italics		y	y	
Line feed option	y	y	y	y
Line spacing	y	y	y	y
Lines per inch	y	y	y	y
Lines per page	y	y	y	y
Linked files	y	y	y	y
Mail merge (variable blocks)	y	y	y	y
Multiple copies	y	y	y	y
Offset	y	y		
Page length ignore (rough draft)			y	
Page numbering	y	y	y	y
Paper length	y	y	y	y
Paragraph indentation		y	y	
Paragraph spacing			y	
Pause	y	y	y	y
Print characters not on keyboard	y	y	y	y
Print selected pages	y	y	y	
Ragged left	y	y		y
Reverse	y			
Right justification	y	y	y	y
Semi-Proportional spacing		y		
Shadow	y			
Superscipt & subscript	y	y	y	y
Supports non–Commodore printers	y	y	y	y
Triple spacing	y	y	y	y
Underlining	y	y	y	y
Vertical positioning	y	y	y	y

TABLE 3: Table of Disk Functions

	Easy Script	Paper Clip	Script 64	Word Pro
Backup	y	y	y	y
Change device number		y		
Collect	y	y		y
Copy	y	y	y	y
Directory	y	y		y
Error messages	y	y		y
Header (format or "new")	y	y	y	y
Initialize	y	y		y
Load prg file	y	y	y	y
Load seq file	y	y	y	
Pattern matching of file names	y	y		y
Rename	y			y
Replace	y	y	y	y
Save prg file	y	y	y	y
Save seq file	y	y		y
Scratch	y	y		y
Select drive number	y	y	y	y
Validate	y	y		y

Conclusions

As was said before, this review is not intended to be competition. Each has its own pros and cons, and I certainly have my prejudices. It is very difficult to give the "feel" of a program in print, and so difficult to make recommendations. I would like to make some general suggestions, however, on whom each program may be most suitable for.

Easy Script uses a few major, non–standard format characters, although it is very similar in feel to WordPro. It is a straightforward program with some nice features, but missing others, such as arithmetic functions. Its documentation is geared for the novice and I also would recommend it for home or personal use.

PaperClip is the most sophisticated of these packages, the most replete with features and thus the most "professional" in feel. It has many unique features and is geared to the writer who does his text creation at the keyboard, rather than copying hand–written notes. There is a significant bug in a common command as discussed earlier. The manual is excellent and suitable for all levels, as is the program.

Script 64 is unique in its approach to handling data in screens. It is strange to compose text at the keyboard and have to remember to manually advance every 22 lines. This screen–orientation does make the handling of certain kinds of data better, and it can function as a free–form database with full word processing capabilities, something unique on the market. On the whole, however, I find the program complicated to use, and although the manual is thorough, it is not well indexed, which makes using the program all the more difficult. It has a significant bug in its printer output that will affect users of the popular Epson line of printers.. It does contain a built–in optional 80–column display, spelling checker and database program which make it quite a bargain.

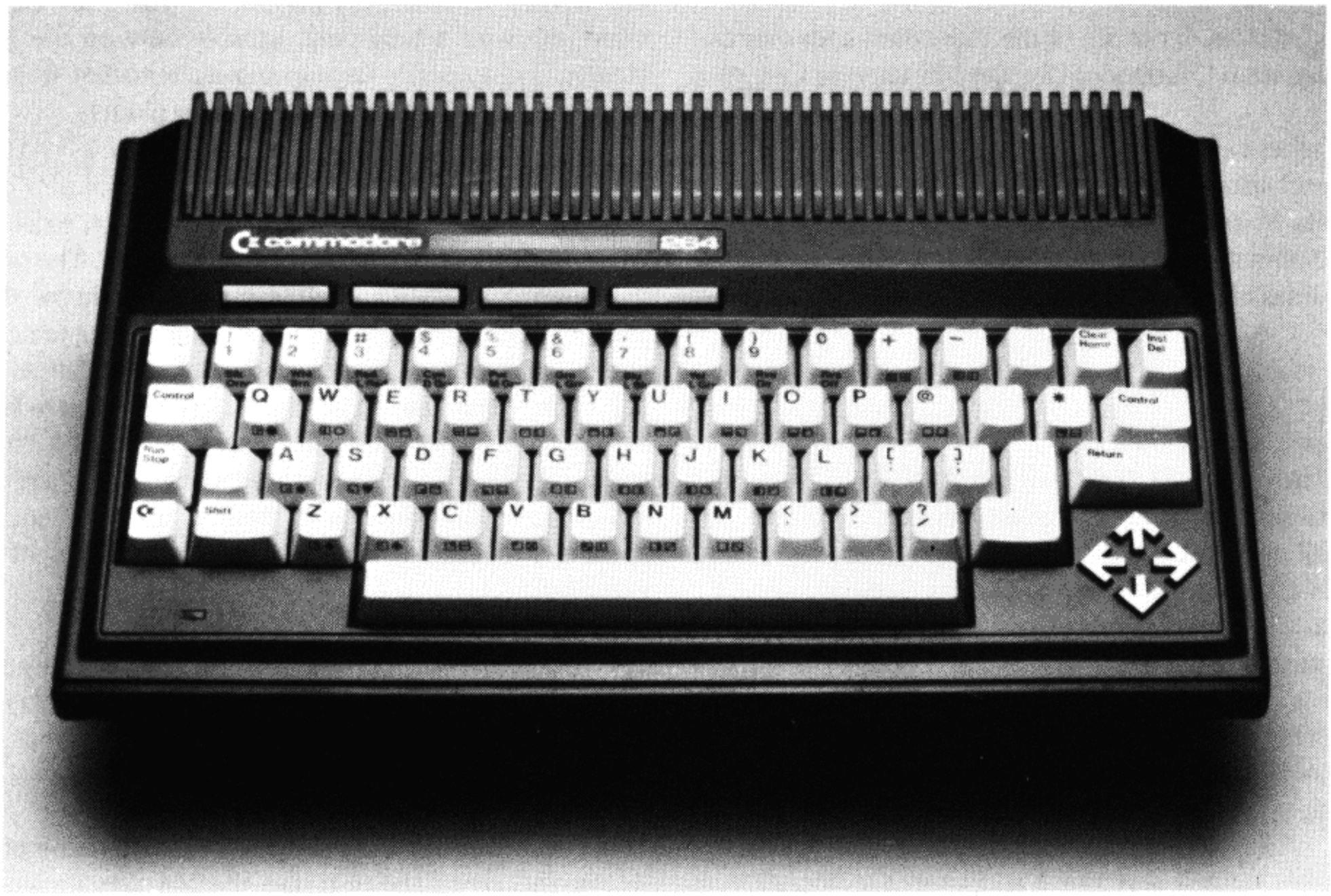
I find WordPro the most straightforward of the programs, but I will acknowledge up front that I have been using one version or another of this program for a few years and so have some biases. It is the easiest to learn, in my opinion, one of the simplest to use, is bug–free and its far–sighted error–trapping has saved my bacon on more than one occasion. It has long been marketed as a business tool, although it was quickly adopted by Commodore users of all types when the first version for the PET came out. It would certainly be perfect for use in an office environment, particularly by people who were more interested in transcribing notes than every fancy feature and consequently having to figure out how the program is supposed to work. Its low price makes it especially attractive, although the inability to back it up makes me more than a little nervous.

It is important when judging any software not to be overawed by all the features any one program may have. Any word processing program intended to be competitive in today's market must have them. Look, in addition, at such qualities as ease of use, error handling and customer support. Hopefully this article has given you enough information to make a happy choice for your particular needs.

The New Commodore Computer

Colin Thompson
Santa Monica, CA

Originally called the "TED", Commodores' new 264 and 364 are their latest entries.



Commodore has finally given us a new computer. The long awaited event took place at the Winter Consumer Electronics Show in early January. The C264 is not an enhanced C64, but a completely new machine geared for the home market. The new 8–bit machine, announced amidst much hoopla and fanfare, will feature "built–in software", on ROM.

Commodore has gone to outside sources for much of the new software for the 264. When the new computer is released this spring, you will be able to buy several versions of the 264, depending on which software package you want "on ROM". The final selection of which programs go on ROM has not yet been made, but likely candidates include:

Word Manager/Plan Manager by Data 20
Magic Desk by Commodore
3–IN–1 by Tri Micro
SuperScript 264 by Commodore
Logo by ??

The idea behind the 264 is simple: If you are going to be using the computer for wordprocessing, then buy a 264 with a built–in wordprocessor. At CES, Commodore brandished the word "Productivity" like a club. New 264 owners should be able to plug in the computer and immediately begin writing letters to the editor on the built–in wordprocessor. But you say, "What if I don't like their W/P?" You will be able to purchase a stripped down 264 with no added ROM programs. What you'll get is a 40 column, colour computer with 60K of available RAM, BASIC 3.5 with over 75 commands, hi–res graphics in 128 colours and a 32K operating system in ROM. But wait, there's more. Did I mention speed? It's faster than the VIC, which is faster than the C64. The 264s new parallel disk drive is about 3–4 times faster than the old 1541. The new computer will not have a SID chip or sprites.

BASIC 3.5 contains several new commands in all areas including editting, program structure, graphics, disk operation, I/O, and machine language. Here's a summary:

Editting:

The 264 has new cursor controls that are arrow shaped keys and seperate from themain keyboard unlike former models.

FIND/CHANGE – search or search & replace BASIC text
AUTO – auto line numbering
RENUMBER
DELETE –10 – delete BASIC lines up to line 10. parameters work like LIST.
TRACE
HELP – hi–lites error in RVS field

Structure:
IF THEN ELSE – all must be on same line
TRAP 1000 – equivalent to ON ERROR GOTO 1000
DO LOOP – can be followed by WHILE or UNTIL
EXIT – terminate loops
GETKEY – instead of 10 GET A$: IF A$ = "" THEN 10

Graphics:
The 264 contains most, if not all, of the VSP commands you can buy on cartridge from Commodore for the VIC and the C64, plus some extras.
FLASH – flashes strings
COLOUR – sets background, multi–colours, etc., AND luminance
GRAPHIC – sets hi–res, multi–colour, or text screen, including combinations. can set hi–res/multi–colour at top plus 5 lines of text screen at bottom.
BOX
CIRCLE
PAINT
SCNCLR – same as print chr$(147)
CHAR – CHAR X, Y, "STRING" will print "STRING" at an X,Y position on the hi–res screen. very handy.
LOCATE – will plot a pixel at X,Y
DRAW – for drawing lines, just like VSP
SSHAPE – Save Shape will store a limited area of the screen into a string variable.
GSHAPE – will get a shape from a specified variable and print it on the hi–res screen at X,Y
SOUND – single voice. followed by parameters for note, tone, etc.
VOLUME
JOY(N) – reads joystick port 1 or 2 and returns a number from 1–8.

Machine Language:
The Machine Language Monitor has some additions as well as some new BASIC commands to do conversions.
DEC – DEC "FFFF" converts the string FFFF to decimal. variable can also be used.
HEX$ – HEX$(1024) converts the number 1024 to a string representing the hexadeciaml equivalent. the two complement much like ASC and CHR$
MLM .F – Fill memory from ADDR1 to ADDR2 with specified hex value
.H – Hunt memrory
.A – Assemble. works like Supermon assembler
.D – Disassemble
.M – Memory dump displays memory contents in hex and screen POKE characters. much like Interrogate in Extramon.
– .G, .X, .S, .L, .R are still the same from earlier MLMs

Disk Operating:
The 264 has most of the disk commands from BASIC 4.0. For some reason, though, the DOPEN command is missing.

(The operating system supports more commands than listed here. We'll bring you more information as release version manuals become available. – Ed.)

Several new peripherals will accompany the 264 including a letter quality and dot matrix printer. A new colour monitor, the 1703, is merely the 1702 in a new black case. This too is the only difference between the new C1542 and the 1541, but a new faster disk drive, the SFS 481 another 264 peripheral. The 1531 Datasette is the same unit with a new plug, useable only on the 264. The 264 Modem is supposed to be auto answer/auto dial. Rounding out the hardware end of the computer is a new plotter.

Software compatibility will be a sore spot with new owners of the 264. VIC and 64 cartridges won't work. Most existing programs won't work either unless they are pure BASIC. The problem lies in the system's electronic architecture. If you know the difference between an operating system and an operating table, you may be wondering how an 8 bit microprocessor can address 64K of RAM, 32K of O/S ROM and an additional 32K of program ROM. The feat is accomplished with a sleight of hand called "bank selected memory". The 7501 Microprocessor can only "see" 64K of anything at once, but with careful memory management, it can switch in (or out) different blocks of memory so the total of 64K is never exceeded.

The bank selection process is done automatically so you don't have to worry about it. Long–winded programmers will now be able to write programs (in BASIC, M.L., or both) of up to 60K in length.

The 264 is the first of four New Generation computers due for release this year.. The next one, the 364, was shown at CES but it looks like it will be summertime before it makes a market appearance. The 364 is architecturally the same as the 264, but has an extra 19 key numeric pad, a few more empty sockets inside for software ROMs, and built–in speech with a 250 word vocabulary.

Prices and release dates are not yet firm, but the 264 should be in the $400–600 US. range.

Editor's Note

The new C264 and 364 are an interesting progression in the Commodore evolution. No sprites, no 3 voice sound, yet very powerful. Perhaps they're leaving that combo for the next step. Whatever the case, the 264 will fill another gap. The added memory, commands and display features will give a new look to Commodore business software, especially the luminance control. It's compact, inexpensive, and ROM software will make it convenient. The Commodore 64 will remain the superior games machine but the SSHAPE; GSHAPE and the other VSP commands will produce some handsome graphic control. It almost seems like Commodore has omitted the advanced graphic and sound capabilities to avoid instilled anxiety in potential business software writers. I suppose only time will tell how well they'll get along together as a family.

Subroutine Eliminators

Jeff Goebel
Acton, Ont.

Hello again, here is my second article in which I expose some little known Commodore features or POKES which have been around for years, but nobody has chosen to discuss them. Last month, I threw in my two cents worth by offering a one line alternative to everyone else's alternative to Commodore's input statement. It was worth buying a back issue if you missed it. (Really! I know they all say that.)

Again this month, I won't let anyone down. I am supplying a one line routine which will eliminate entire routines. First, let's set up a scenario. Your computer is sorting 1000 names using a simple shell sort. (Last month's example used a bubble sort but I graduated.) Your computer is going to be busy for some time and you just don't have the time to sit in front of the screen waiting. What you need is a method for your computer to tell you "I'M DONE!". Obviously the built in CB2 sound is the answer, but you really don't have the time to program even a simple beep. You have to set up the right values in the right locations and experiment till you get a nice tone, when what you are really interested in is the results of your sort and what to do next.

There is the problem, now for the answer. Ideally, it would be nice to get a nice "DING–A–LING–A–LING" just like the one you get when you turn on the computer, right? You can, without knowing anything about programming. All you have to be able to do is print a reverse field G. It is simple to do. Type in PRINT, followed by two sets of quotation marks. Then, delete the second set. Now touch the RVS key. If you did everything right, a reverse R should not appear. Now simply touch the G key, or the Shifted G for a little longer beep. When you hit return, you should here a beep. If you want a longer tone, type several G's together. Couldn't be easier.

I can go one step further if you own a Commodore 8032 or SP9000. First, you need not bother with the second set of quote marks. Touching the ESC (Escape) key ,takes you out of quote mode – hit RVS to engage reverse field characters and type the G or shifted G key. Further, on these computers you can control the length of the beep. Try the above routine. Then POKE 231,255 and try it again. This time the tone will take a much longer time to finish. Experiment with values from 0 to 255. For example:

```
10 for j=0 to 255
20 poke 231, j
30 print j; " G "
40 next
```

The location 231 on the 4032 creates a really bizarre side effect, much like the sort of thing Karl or Richard might publish in their "Bits And Pieces" section. It creates sort of an anti–window with text. Experiment if you like, but if it wrecks your computer, I didn't tell you. Just kidding of course... simply power down to get back to normal.

One last note on the subject... the concept of printing a reverse G is sometimes referred to as "Control–G". Sending a Control G is virtually an industry standard among computers that host some kind of beeper at the keyboard. So if you want to write software that is transportable between different brands of computers, Control G is a fairly safe way to create a beep for the user, even while telecommunicating.

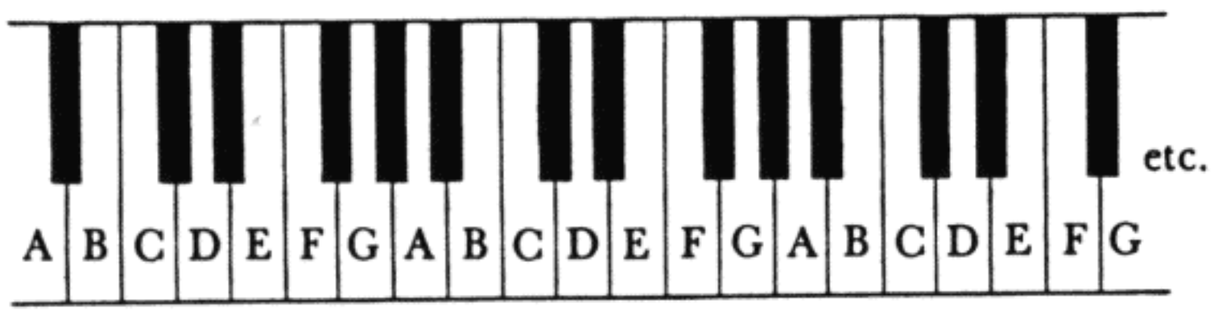
A Quick Musical Tour

With this issues' theme of graphics and sound, what better occasion could be found to write a quick tour of musical concepts. Most people, at one time in their life or another, have had at least a small urge to know a bit more about music and the concepts behind it. Everyone who has ever debated doing anything as major as designing a music synthesis system, or as minor as trying to hear their computer make some form of noise, have felt it in their best interests to become slightly knowledgeable in the concepts of music. Well, sit back and enjoy. The reading that you are about to do has been written with you in mind. No mind boggling terms will be stressed, just basic concepts to whet your musical appetite sufficiently for future research. So, with that introduction out of the way, I welcome you to a quick guided tour through the concepts of music.

You have been transported to ESP studios, and are now sitting behind a beautiful ebony Grand Piano that has been gratuitously donated for our use today. A quick look around the studio will tell you that the owners are in the business of making music and do very well in their occupation. Some of todays top name groups have passed through this studio on their way to musical millions, and now you can join their ranks, if you can just play the piano you sit behind.

Cautiously peer down at the keyboard and try to get a bearing of how everything fits into place. You will notice that the white keys fall in exact sequence from the far left of the keyboard to the far right hand side. The black keys, in contrast, seem to appear between most of the white keys, but in a seemingly irregular pattern.

What has been discovered is the first in a series of very important points. Each white key on the keyboard is referred to as a **WHOLE NOTE**, with the black keys being the **FLAT's** and **SHARP's** of their respective WHOLE NOTEs. The keys to your left are referred to as the **BASS** or low notes, with the keys to your right being the **TREBLE** or high notes. Now, just in case everyone reading did not manage to get a seat at the piano, a small diagram of the keyboard can be found below.



As the diagram above shows, each white key has a note letter name attached to it with the names ranging in sequence from A to G, then back to A again. What you have witnessed is the range of notes, A through G, in numerous **OCTAVES**. When two notes are one OCTAVE apart, this means that they have the same letter name but are either seven **WHOLE NOTEs** higher or lower than the other. If you were playing a C note on the far left side of the keyboard and were asked to raise the note one OCTAVE, you would leave that key and move seven white keys to your right to land on the next C note one OCTAVE higher. Pretty easy concept to grasp, so far.

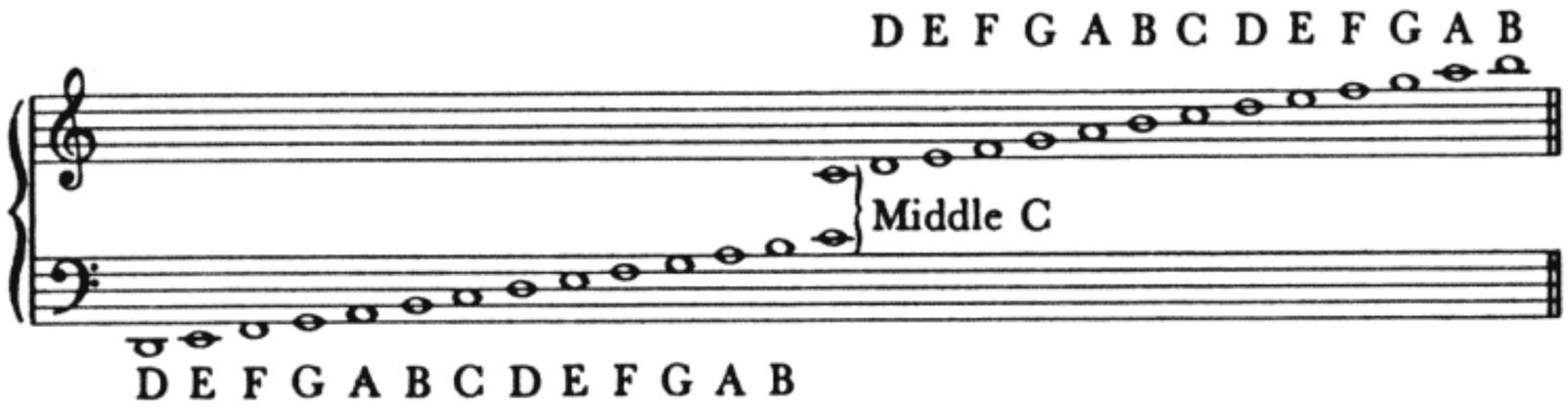
As you play on your new C key, one octave higher than before, a call comes from the control room to play a C SHARP for a few moments. Don't panic, a SHARP or FLAT, as mentioned earlier, are the black keys that fall between the white ones. A SHARP is usually the black keys that fall to the right of most of the white keys, with a FLAT usually being the black keys to the left of the white keys. The phrase, *usually*, will be explained further in a short while. If you were to do as instructed and played the C SHARP, you would notice that the tone is slightly higher than the C that you were playing. This increase in tone is called a **SEMI-TONE**, or one half of a WHOLE TONE. Press the white key just to the right of the C key for a second, and you will notice another slight increase in tone, or a raise of another semitone to become a **WHOLE TONE**. This is the D note.

The explanation given above is sure to relieve numerous questions in your mind right now but a very large one should remain. Earlier you have discovered that the sequence of black keys in relation to the white ones are in a rather irregular order, and you also picked up on the fact that I said *usually* when explaining the FLATs and SHARPs to you. This is due to the fact that, when the keyboard was originally conceived, a few black keys were left out for some purpose I have never been able to figure out. This should have lead to a large musical void in composition, but it hasn't. This drawback has not appeared to hinder one composer through time, so it shouldn't affect us now.

The keys affected by this strange turn of events are the B, C, E & F keys. To compensate for this loss, the white keys are used in place of the blacks. For example, a B SHARP is a C, with a C FLAT being a B. An E SHARP is a F, with a F FLAT being an E. Strange but true, and a musical fact of life for all time to come.

Now, if you were to look up from the keyboard for a few moments, you will notice that Dee, of ESP, has placed some sheet music in front of you. Those squiggles, lines and strange markings all have special meaning, and each deserve a quick explanation at least. Though sheet music may seem Greek to you right now, just think of the poor musician trying to decipher an assembler listing or BASIC program. The confusion level would be just as high, so sit back and try to at least visualize some of this.

As explained earlier, notes run through cycles of A through G numerous times along the keyboard in assorted **OCTAVES**. At the far left end of the keyboard, you will notice that the first key is a white C note. The next WHOLE NOTE is a D, with the trend continuing on accordingly. At the far right side of the keyboard, the keys terminate with another C note. The entire scale on the ESP Grand Piano is from C to C. Other keyboards that you run across do not cover this entire range, but we are fortunate today to be exposed to a truly fine piano.



Above is a diagram to help you understand the placement of these notes on the written scale. To begin, a **STAFF** is the five parallel lines, and the four spaces between them, on which notes and other musical characters are placed. In the diagram above will be noted two complete STAFFs, the **BASS** and **TREBLE** STAFFs. The treble staff can be identified on the diagram, as the one starting with the G shaped character at the far left beginning of the staff, which is the **TREBLE CLEF**. Below that will be noticed a really funny looking 9 shaped character, which is the **BASS CLEF**. This signifies the bass staff. In easier terms, the bass staff pertains to the left side of the keyboard, with the treble staff relating to the right hand side of the keyboard. In the middle of the **SCALE** in the diagram above falls the **MIDDLE C**, or a note that has been placed in the very centre of the keyboard. The left or right hand side of the keyboard is always assumed to be relative to the position of middle C.

On the diagram, each note is represented by a circular, hollow dot or a **WHOLE NOTE**. The true names for these notes will follow at the end of this article for those interested. This description refers the **DURATION** of the note in question. One half of a **WHOLE NOTE**, predictably, is a **HALF NOTE**. Below that is the **QUARTER NOTE**, then **EIGHTH NOTE**, and so on. Each of these notes have a corresponding character attached to it, with the **HALF NOTE** looking like a **WHOLE NOTE** with a tail attached. The **QUARTER NOTE** looks like a **HALF NOTE** with the circular hollow dot filled in. An **EIGHTH NOTE** looks like a **QUARTER NOTE** with a further tail projecting from the tip of its tail at an approximate 135 degree angle. Each of the notes beyond this point have further variations, with enough difference to be quickly distinguishable. Hopefully the diagram below will help you a bit more in visualizing this.

NAME	SIGN	VALUE COMPARED TO 𝅝	REST
SEMIBREVE	𝅝		𝄻
MINIM	𝅗𝅥	1/2 𝅝	𝄼
CROTCHET	♩	1/4 𝅝	𝄽 or ♩
QUAVER	♪	1/8 𝅝	𝄾

While looking through the sheets of music in front of you, you will notice that the tails of the notes seem to be going both up and down in relation to the the circular dots. This is done to keep the tails from projecting beyond the edge of the staff, to help keep the scores from looking terribly messy. The general rule of thumb is to point the tails upward if the note is below the centre of the staff, and point the tail downward when the note is above the centre. The note on the centre line of the staff can be written as you deem convenient. Truthfully, this is only for esthetic purposes, but it does make for some very readable compositions.

When the FLAT of a note is needed in a piece of music, a very funny looking 'b' is placed before the note corresponding on the staff. For a SHARP, a # is placed before the actual note. To cancel a flat or sharp, a **NATURAL** is placed on the **STAFF** before the note it is to affect. A NATURAL sign looks like a box with two tails extending vertically from the upper left and lower right corners. The FLAT, SHARP and NATURAL signs are referred to as **ACCIDENTALS**, and only affect the note in the same OCTAVE in which it has been written. They do not affect the same note in the other octaves unless they too have been labeled with an ACCIDENTAL. This is why a NATURAL is needed, just in case you happen to need the same note again in the same octave, but without any variation in tone.

Music in general is always written with a certain throb or pulse to it. These pulses are referred to as **BEATS** with the grouping of BEATS being determined by the **ACCENTS** some notes have on them in relation to others. This ACCENT is shown on musical scores as a > sign above the note on the staff. The strongest of the ACCENTS are highlighted by putting a vertical line in front of the note affected. These vertical lines divide the music into equal measures, and are referred to as **BARS**. The end of a piece of music is marked by two vertical lines, or a **DOUBLE BAR**.

The regular grouping of BEATs into BARS is called the **TIME** of the music with the kind of TIME being determined by whether the ACCENTS occur in twos, threes or fours as indicated by the **TIME SIGNATURE** written on the staff after the clef at the beginning of the piece.

If the BEATS fall into regular groups of two, this is said to be in **DUPLE TIME** and a number two is used as the top value in the **TIME SIGNATURE**. Predictably, if the BEATS fall into a sequence of regular groups of three, this is referred to as **TRIPLE TIME** with the TIME SIGNATURE adjusted accordingly. A sequence of four is called **QUADRUPLE TIME** and is marked with a four as the top value on the TIME SIGNATURE. With this out of the way, we now know that the top value on the TIME SIGNATURE refers to the number of BEATS in a BAR.

The bottom value in the TIME SIGNATURE refers to the length of the notes in each BEAT expressed in fractions of a WHOLE NOTE. Therefore, a TIME SIGNATURE of 3/4 indicates that there are three QUARTER NOTE BEATS in a BAR, in TRIPLE TIME. If we had a TIME SIGNATURE of 4/8 then this would mean four EIGHTH NOTE BEATS per bar in QUADRUPLE TIME. You will find that the more you think about this one, the easier it is to comprehend.

Sometimes the TIME SIGNATURE appears to be missing in action, and in its place appears another strange character – a TIME SIGNATURE as in dollars and cents, instead of the 2/2. There are other instances of this occurring, but I feel it best that you take the initiative to find out more about it if you feel so inclined.

Within beats you will most likely notice note signs that did not seem to appear in my earlier description. They are all tied together at their tails, totally unlike anything described earlier, but they are really not so strange. The tieing together of the tails of EIGHTH NOTES is common practice within BEATS, just for the purpose of esthetics once again. Music is really a very tidy business.

Quite often while composing music, a **REST** between notes is desired to break up the music a bit, and provide pleasurable listening instead of a continuous stream of notes. If a space was left on the staff, the person playing the piece would not really know what was meant. The only way to express the REST properly would be with special symbols once again. In the note chart shown earlier, you have probably noticed the far right column of strange symbols under the heading**REST**. Those are the characters to look for while playing music from sheet. The corresponding note durations are also shown to the far left.

To conclude this article, I would like to bring to your attention all the key points covered today.

Octave –Twelve semitones or all notes between A through G on the keyboard, black and white keys.

Bass Clef –The 9 shaped character symbolizing the BASS STAFF

Treble Clef –The G shaped character indicating the TREBLE STAFF

Semitone –One half of a tone, or the distance between two adjacent notes.

Whole Note –Two semitones.

Sharp –A raising of one semitone of a tone.

Flat –A lowering of one semitone of a tone.

Natural –A character used on sheet music to return a note to its normal value.

Accidentals –What the FLATs, SHARPs, and NATURALs mean in relation to each other: They only affect the corresponding notes within the OCTAVE in which it is applied.

Whole Note –**(SEMIBREVE)** a duration of one whole note.

Half Note –**(MINUM)** a duration of half of a whole note.

Quarter Note –**(CROTCHET)** a duration of one quarter of a whole note.

Eighth Note –**(QUAVER)** a duration of one eighth of a whole note.

Beats –The pulse or throb of music when played.

Bar Lines –Vertical line used in front of the most heavily stressed accents of the music.

Bars –Divide the music into equal measures.

Double Bars –To signify the end of a musical piece.

Time –The regular grouping of beats into bars.

Time Signature –The kind of time, either in twos, threes or fours.

Duple Time –The beats fall into regular groups of two.

Triple Time –The beats fall into regular groups of three.

Quadruple Time –The beats fall into regular groups of four.

Alla Breve –Another name for a time signature of 2/2, which means two half note beats in a bar in duple time.

If, after you have completed this article, you have an unsatiable appetite for more, I recommend the nearest library or bookstore. If your appetite is still there after all that trudging about, send us a letter and a further musical tour may appear again. Whatever the story, I hope that you enjoyed the short time that we have spent together. As a final note, I would like to thank John, Dee and the rest of the guys at ESP for the loan of their piano. Without it, this article may never have appeared. RTE

WAVES For The 64

Chris Zamara
Downsview, Ont.

Sooth your brain and be more productive!

An interesting psychological phenomenon is the soothing effect which the sound of the ocean's waves seems to have on people. It seems that the rythmic rush of white noise makes one feel more relaxed, and some people find that having a white noise source (or the tape–recorded sound of breaking waves) in the room that they are working makes them more productive.

If you sit behind your computer programmming for hours at a time, and you start going nuts after a while, try running WAVES before your session. Once WAVES is enabled, it runs during the interrupts, and simulates the sound of waves breaking on a shore. You still have complete control of your computer, so you can program with the sound of waves in the background. Set the volume control on your TV or monitor to any level you find comfortable, and try it for a while. It might really help you think (On the other hand, it might just drive you even more nuts, but it's worth a try).

If you don't have volume control, the number 15 shown in bold on line 280 is the volume value. 15 is maximum, but if you lower it, you will also have to lower the checksum test value in line 190 by the same amount or an error will be reported. The easiest way to disable Waves completely is simply change the master volume to 0 with:

```
POKE 54296, 0
```

Then, at any time, you can bring Waves back by POKing 54296 with some volume value from 1 to 15. That's assuming the interrupt vector hasn't been changed by some other program.

Enter the BASIC program below and RUN it. If you get a checksum error, check the DATA statements. Once the checksum is OK, WAVES will be enabled. It takes a few seconds for the first wave to start. Some technical notes: Voice 1 in the SID chip is used for the sound, and you can vary the frequency, attack/decay, and sustain/release parameters of that voice to get different sounding waves. The word at 49193 and 49194 (low, high) controls the number of jiffies (1/60 seconds) that the voice is gated, and the word at 49195 and 49196 controls the ungated time.

```
100 rem * data loader for waves *
110 :
120 cs=0     : rem * checksum    *
130 os=49152  : rem * object start *
140 :
150 read b: if b<0 then 180
160 cs=cs+b
170 poke os,b: os=os+1: goto150
180 :
190 if cs<>16560 then print " *** checksum error *** "
      : goto 240
200 :
210 sys 49152 : rem * enable waves
220 print " ** Ok, WAVES is enabled. ** "
230 :
240 end
250 data 169, 168, 141,    0, 212, 169
260 data  97, 141,    1, 212, 169, 221
270 data 141,    5, 212, 169, 125, 141
280 data    6, 212, 169,  15, 141,  24
290 data 212, 120, 169,  45, 141,  20
300 data    3, 169, 192, 141,  21,   3
310 data  88,  96,  88,    0, 255, 200
320 data    0, 144,    1, 238,  38, 192
330 data 208,    3, 238,  39, 192, 173
340 data  40, 192, 208,  37, 173,  38
350 data 192, 205,  43, 192, 208,  63
360 data 173,  39, 192, 205,  44, 192
370 data 208,  55, 169, 129, 141,   4
380 data 212, 169,    0, 141,  38, 192
390 data 141,  39, 192, 169, 255, 141
400 data  40, 192,  76, 129, 192, 173
410 data  38, 192, 205,  41, 192, 208
420 data  26, 173,  39, 192, 205,  42
430 data 192, 208,  18, 169, 128, 141
440 data    4, 212, 169,    0, 141,  38
450 data 192, 141,  39, 192, 169,   0
460 data 141,  40, 192,  76,  49, 234
470 data  -1
```

Programming Sound On The VIC 20

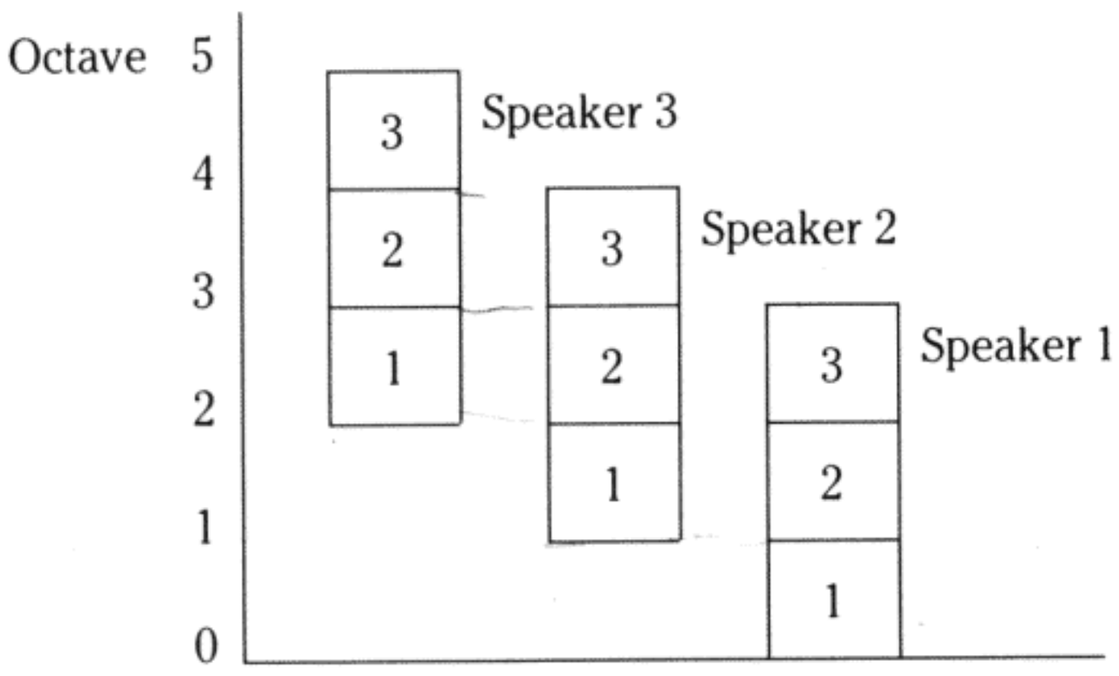
Dave Gzik
Burlington, Ont.

By now, all you VIC 20 users have heard sound produced on your machine. For some, you have probably experienced more with the sound capabilities than others. In the following article I hope to show you some neat tricks with effects, and musical reproduction.

First of all lets show you what makes up the sound system in the machine. Basically you will only affect 5 locations in the VIC, and they are:

Location	Description
36878	Volume
36877	Speaker 4 (noise)
36876	Speaker 3 (high)
36875	Speaker 2 (middle)
36874	Speaker 1 (low)

Each speaker can play in three different octaves and the higher the speaker number, the higher the notes. To show this to you graphically, it would look like this:



Notice that each speaker has some sort of redundancy of notes. Instead of 9 full octaves, the range from the lowest note capable to the highest note capable is 5 octaves. This overlapping allows the blending of the same note for a richer tone.

Speaker 4 is a noise generator which can be used for sound effects. The volume setting can be in the range from 0 to 15.

The values used for musical notes in all speakers are in the range of 128 to 255 including speaker 4. A table for note values is shown on the next page.

That's it for the description of the system, now lets put it to work.

Basically, to play a note you set the volume, turn on the note, play it for a while, and then turn it off.

To play it for a while is called the duration. The easiest way to accomplish the duration is with a FOR/NEXT loop.

Try this example:

```
10 vol=36878:s1=36874:s2=36875:s3=36876:s4=36877
20 poke vol,15
30 poke s1,200
40 for dur=1 to 250 : next
50 poke s1,0
```

This program sets the volume to 15, then puts the note value of 200 into speaker 1, and plays it for the duration count of 250. When the duration is finished it turns the speaker off. This is the standard routine, and you can expand it for use with multiple speakers and different durations for the notes.

To recap, the basic program flow is:

1 Set variables for volume and speakers.
2 Set volume setting.
3 Turn on speaker.
4 Play it for a while.
5 Turn it off.

To play music may require the use of data statements for the notes. The best way to set them up for ease is in this format:

DATA speaker1 note,speaker2 note,speaker3 note,duration

The corresponding read statement would then look like this:

READ S1, S2, S3, DUR

Now the basic program structure changes to this:

1 Set variables.
2 Set volume.
3 Read notes, duration from data.
4 If data is completed then exit program.
5 Turn on speaker.
6 Play it for a while.
7 Turn it off.
8 Go back to step 3
9 End reset all variables to zero
10 Data elements.

Below is a full fledged piece of music that puts the precreding logic to work. Study it carefully to see how it is structured.

Next issue I will go further into manipulating the previous ideas for special sound effects, etc.

I leave you with a section of BACH'S invention number 4. Have fun.

```
10 vol=36878 : s1=36874 : s2=36875 : s3=36876
20 poke vol,5
30 read x, y, dur
40 if x=-1 then 80
50 poke s2,x : poke s3,y
60 for t=1 to dur*18 : next
70 goto 30
80 poke vol,0 : poke s1,0 : poke s2,0 : poke s3,0
90 end
1000 data   0, 201,   8,   0, 207,   8,   0, 209
1010 data   8,   0, 215,   8,   0, 219,   8,   0
1020 data 221,   8,   0, 199,   8,   0, 221,   8
1030 data   0, 219,   8,   0, 215,   8,   0, 209
1040 data   8,   0, 207,   8, 201, 209,   8, 207
1050 data   0,   8, 209, 219,   8, 215,   0,   8
1060 data 219, 228,   8, 221,   0,   8, 199, 215
1070 data   8, 221,   0,   8, 219, 227,   8, 215
1080 data   0,   8, 209, 231,   8, 207,   0,   8
1090 data 209, 228,   8,   0, 231,   8, 219, 232
1100 data   8,   0, 235,   8, 228, 237,   8,   0
1110 data 238,   8, 207, 227,   8,   0, 238,   8
1120 data 215, 237,   8,   0, 235,   8, 227, 232
1130 data   8,   0, 231,   8, 201, 232,   8, 201
1140 data 228,   8, 228, 231,   8, 228, 232,   8
1150 data 209, 235,   8, 209, 237,   8, 215, 221
1160 data   8, 215, 237,   8, 219, 235,   8, 219
1170 data 232,   8, 221, 231,   8, 221, 228,   8
1180 data 195, 231,   8, 195, 225,   8, 225, 228
1190 data   8, 225, 231,   8, 207, 232,   8, 207
1200 data 235,   8, 209, 219,   8, 209, 235,   8
1210 data 215, 232,   8, 215, 231,   8, 219, 228
1220 data   8, 219, 225,   8, 201, 209,   8, 187
1230 data 215,   6, 187,   0,   4, 195, 215,   4
1240 data 195, 219,   4,   0, 215,   4, 135, 215
1250 data   8, 135, 209,   6, 135,   0,   8, 163
1260 data 209,   8, 175, 209,  32
5000 data  -1,  -1,  -1
```

VIC 20 Note Values

Where two values are shown, it is necessary to alternate between them to get the true note.
Voice frequency registers are 36874/5/6. Noise reg is 36877. Volume is Lo nybble of 36878. See Memory Map

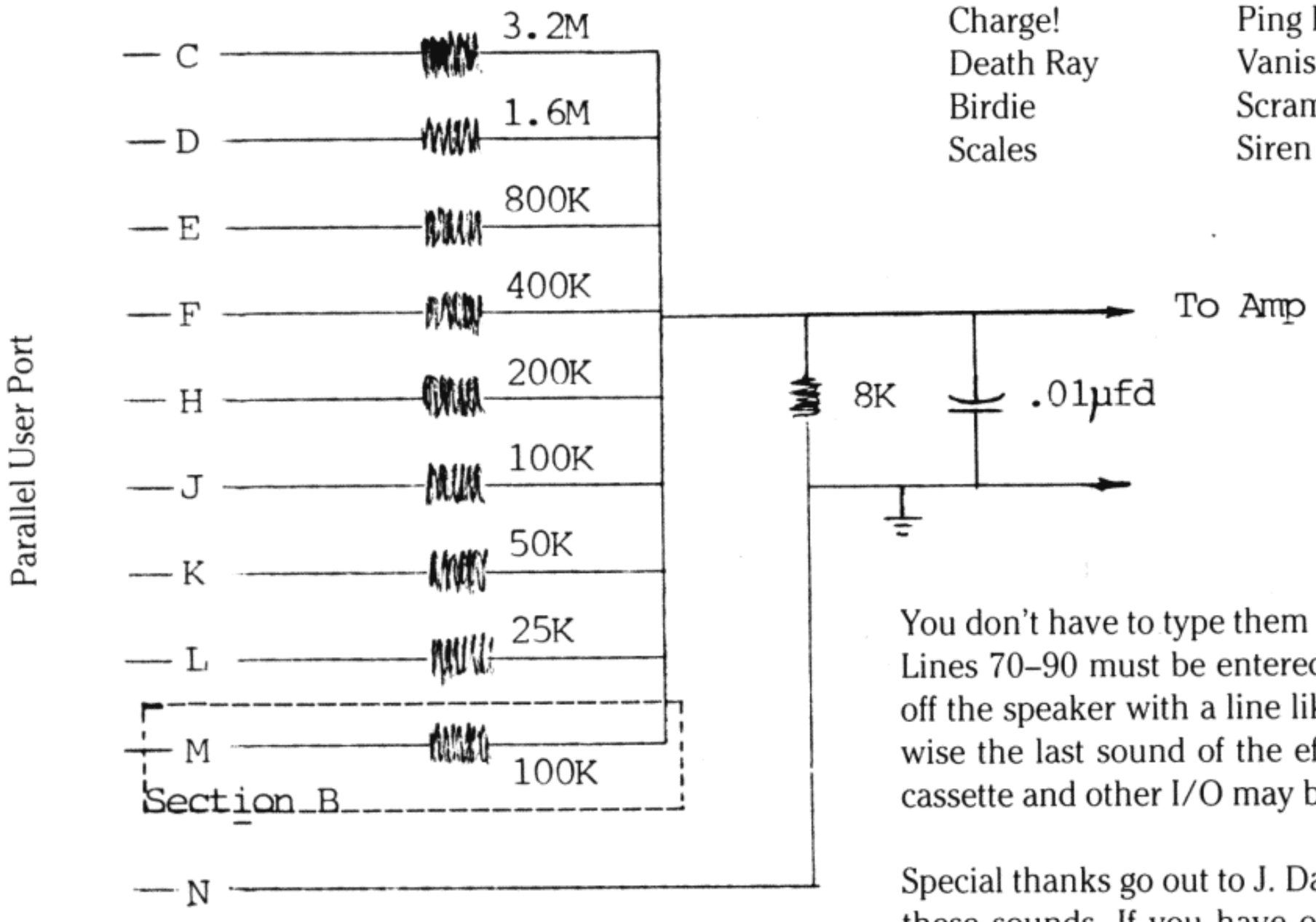
Note	Octave 0		Octave 1		Octave 2		Octave 3	
	Value	Mod. Val.	Value	Mod. Val.	Value	Mod. Val.	Value	Mod. Val.
C	131		192	195	224		239	240
C#	140		197		226		240	241
D	145		200		227	228		
D#	151		203		229			
E	158		206	207	231			
F	161	162	208	209	232			
F#	166	167	211	212	233			
G	173	174	214		234	235		
G#	178		216		238	236		
A	181	182	218	219	237			
A#	185	186	220	221	237	238		
B	189	190	222	223	239			

Sound Effects

Programming sound effects on any Commodore machine requires but two things: the knowledge of how it works and a vivid imagination. If you only one of the two, this next listing will give you the other.

The sound effects presented here were originally written for the PET/CBM. Late model PET/CBMs had a built–in beeper that you won't find in earlier models, the VIC 20 or the Commodore 64. But that's no reason why they won't work on these machines too. What you need is a "Digital to Analog Converter". Many firms have marketed D/As in the past, but they're ridiculously simple to make. Below is a circuit schematic of a passive D/A converter.

With the circuit in place, connect the output to the auxiliary input of any stereo system. It's probably best to use an RCA male phono jack.



The corresponding chip locations for the VIC and C64 are:

PET/CBM	VIC 20	Commodore 64
59464	37144	56582
59466	37146	56584
59467	37147	56585

If you're not inclined to build this item, it shouldn't take terribly long to convert the program to work with the internal sound already provided for in the VIC 20 or Commodore 64. For clues, take a look at VIC Sound by Dave Gzik in this issue, or Zoundz for the C64 by Howard Strasberg in the Bits and Pieces section.

The sound effects you will hear are:

Piano Gliss	Laser Fire	French Police
Bombs Away	Wolf Whistle	Dive! Dive!
Shave 'n' a Haircut	Dirge	R2D2
Charge!	Ping Pong	Raspberry
Death Ray	Vanishing Saucer	Eat My Dust
Birdie	Scramble	Bounces
Scales	Siren	Ticking Clock

You don't have to type them all in – just the ones you want. Lines 70–90 must be entered and it's good practice to turn off the speaker with a line like 160 after each effect. Otherwise the last sound of the effect will remain on, plus your cassette and other I/O may be adversely affected.

Special thanks go out to J. David of WHCI for work done on these sounds. If you have others, send them in and we'll publish another collection like this in a future issue.

```
70 tt=59464 : rem timer 2 low byte
80 sr=59466 : rem shift register
90 sc=59467 : rem shift register control
100 print " *** piano gliss *** "
110 poke sc,16 : poke tt,0 : poke sr,15
120 for k=0 to 100 step5
130 poke tt,k : x=tan(k) : next
140 for k=99 to 0 step-5
150 poke tt,k : x=tan(k) : next
160 poke sc,0 : poke sr,0 : poke tt,0
170 rem return
180 print " *** laser fire *** "
190 poke sc,16 : poke tt,0
200 poke sr,15 : for k=1 to 5
210 for r=0 to 100 step5
220 poke tt,r : for x=1 to 2 : next : next
230 poke tt,r : x=tan(x) : next
240 poke sc,0 : poke sr,0 : poke tt,0
250 rem return
260 print " *** french police *** "
270 poke sc,16 : poke tt,0 : poke sr,2
280 for k=1 to 4
290 for r=100 to 255 step40
300 poke tt,r : for x=1 to 500 : next
310 for r=255 to 100 step-40
320 poke tt,r : next : for x=1 to 500 : next
330 next
340 poke sc,0 : poke sr,0 : poke tt,0
350 rem return
360 print " *** bombs away *** "
370 poke sc,16 : poke tt,0 : poke sr,85
380 for r=50 to 150 : poke tt,r : for k=1 to 30 : next : next
390 poke sr,1 : poke tt,255 : for k=1 to 800 : next
400 poke sc,0 : poke sr,0 : poke tt,0
410 rem return
420 print " *** wolf whistle *** "
430 poke sc,16 : poke tt,0 : poke sr,13
440 for r=185 to 80 step-3 : poke tt,r : next
450 poke tt,0 : for k=1 to 200 : next
460 for r=205 to 105 step-3 : poke tt,r : next
470 for r=105 to 255 step3 : poke tt,r : next
480 poke sc,0 : poke sr,0 : poke tt,0
490 rem return
500 print " *** dive! dive! *** "
510 poke sc,16 : poke sr,9
520 for k=1 to 10
530 for r=250 to 180 step-1
540 poke tt,r : next : next
550 poke sc,0 : poke sr,0 : poke tt,0
560 rem return
570 print " *** shave 'n' a haircut, 2 bits *** "
580 poke sc,16 : poke tt,0 : poke sr,15 : t=3
590 poke tt,188 : for k=1 to 200 : next
600 poke tt,251 : for k=1 to 100 : next
610 poke tt,0 : for k=1 to t : next
620 poke tt,251 : for k=1 to 100 : next
630 poke tt,225 : for k=1 to 200 : next
640 poke tt,251 : for k=1 to 300 : next
650 poke tt,0 : for k=1 to 150 : next
660 poke tt,199 : for k=1 to 200 : next : t=50
670 poke tt,0 : for k=1 to t : next
680 poke tt,188 : for k=1 to 150 : next
690 poke sc,0 : poke sr,0 : poke tt,0
700 rem return
710 print " *** dirge *** "
720 poke sc,16 : poke tt,0 : gosub 750
730 poke sc,0 : poke sr,0 : poke tt,0
740 goto 890 : return
750 rem dirge
760 poke sr,15 : t=3
770 poke tt,237 : for k=1 to 300*t : next : gosub 880
780 poke tt,237 : for k=1 to 200*t : next : gosub 880
790 poke tt,237 : for k=1 to 100*t : next : gosub 880
800 poke tt,237 : for k=1 to 300*t : next : gosub 880
810 poke tt,199 : for k=1 to 300*t : next : gosub 880
820 poke tt,211 : for k=1 to 200*t : next : gosub 880
830 poke tt,237 : for k=1 to 100*t : next : gosub 880
840 poke tt,237 : for k=1 to 200*t : next : gosub 880
850 poke tt,251 : for k=1 to 100*t : next : gosub 880
860 poke tt,237 : for k=1 to 300*t : next
870 poke tt,0 : for k=1 to 5 : next : return
880 poke tt,0 : for k=1 to t : next : return
890 print " *** r2d2 *** "
900 poke sc,16 : poke tt,0 : poke sr,15
910 for k=1 to 30 : poke tt,10+100*rnd(1)
920 for i=1 to 6 : next : next
930 poke sc,0 : poke sr,0 : poke tt,0
940 rem return
950 print " *** charge! *** "
960 poke sc,16 : poke tt,0 : gosub 990
970 poke sc,0 : poke sr,0 : poke tt,0
980 goto 1080 : rem return
990 poke sr,15 : t=3
1000 poke tt,255 : for k=1 to 100 : next : gosub 1070
1010 poke tt,191 : for k=1 to 100 : next : gosub 1070
1020 poke tt,152 : for k=1 to 100 : next : gosub 1070
1030 poke tt,128 : for k=1 to 200 : next : gosub 1070
1040 poke tt,152 : for k=1 to 100 : next : t=0
     : gosub 1070
1050 poke tt,128 : for k=1 to 400 : next
1060 return
1070 poke tt,0 : for k=1 to t : next : return
```

```
1080 print " *** ping pong *** "
1090 poke sc,16 : poke tt,0 : poke sr,15
1100 for j = 1 to 5
1110 poke tt,255 : for k = 1 to 60 : next
1120 poke tt,0 : for k = 1 to 100↑rnd(1)*20 : next
1130 poke tt,128 : for k = 1 to 60 : next
1140 poke tt,0 : for k = 1 to 100↑rnd(1)*20 : next
1150 for x = 1 to 100 : next : next
1160 poke sr,63 : poke tt,255 : for k = 1 to 500 : next
1170 poke sc,0 : poke sr,0 : poke tt,0
1180 rem return
1190 print " *** raspberry *** "
1200 poke sc,16 : poke tt,0 : poke sr,9
1210 for k = 1 to 50 : poke tt,238 : poke tt,251 : next
1220 poke sc,0 : poke sr,0 : poke tt,0
1230 rem return
1240 print " *** death ray *** "
1250 poke sc,16 : poke tt,0 : poke sr,15
1260 for k = 1 to 200 : poke tt,150 : poke tt,200
     : poke tt,255 : next
1270 poke sc,0 : poke sr,0 : poke tt,0
1280 rem return
1290 print " *** vanishing saucer *** "
1300 poke sc,16 : poke tt,0 : poke sr,29
1310 for k = 160 to 0 step-.3 : poke tt,k : poke tt,k + 5
     : poke tt,k + 15 : next
1320 poke sc,0 : poke sr,0 : poke tt,0
1330 rem  return
1340 print " *** eat my dust *** "
1350 poke sc,16 : poke tt,0 : poke sr,3
1360 for k = 200 to 235 step.7 : poke tt,k : poke tt,k + 5
     : poke tt,k + 20
1370 for z = 1 to 20 : next : next
1380 for k = 1 to 300 : poke sc,0 : poke sr,16 : poke sc,16
     : poke sr,3 : next
1390 for k = 235 to 170 step-.6 : poke tt,k : poke tt,k + 5
     : poke tt,k + 15 : next
1400 for k = 170 to 220 step3 : poke tt,k : next
1410 for k = 220 to 129 step-.5 : poke tt,k : poke tt,k + 5
     : poke tt,k + 15 : next
1420 for k = 120 to 180 step3 : poke tt,k : next
1430 for k = 140 to 200 step2 : poke tt,k : poke tt,k + 5
     : poke tt,k + 15 : next
1440 k = 200 : poke sc,16 : poke sr,51
1450 for x = 1 to 50 : poke tt,k : poke tt,k + 7
     : poke tt,k + 14 : next
1460 poke sc,0 : poke sr,0 : poke tt,0
1470 rem return
1480 print " *** birdie *** "
1490 poke sc,16 : poke tt,0 : poke sr,85
1500 for k = 1 to 10
1510 for k1 = 152 to 56 step-8 : poke tt,k1 : next
1520 poke tt,0 : for k1 = 1 to int(rnd(1)*200) : next : next
1530 poke sc,0 : poke sr,0 : poke tt,0
1540 rem return
1550 print " *** scramble *** "
1560 poke sc,16 : poke tt,0 : poke sr,85
1570 for k = 1 to 20
1580 for k1 = 1 to 14 : poke tt,k1*16 : next : next
1590 poke sc,0 : poke sr,0 : poke tt,0
1600 rem return
1610 print " *** bounces *** "
1620 for n = 255 to 20 step-5 : poke sc,16 : poke sr,15
     : poke tt,n : poke sc,0
1630 for nn = 1 to 100 : next : next : poke sc,0
1640 rem return
1650 print " *** musical scales *** "
1660 gosub 1670 : goto 1850 : rem return
1670 gosub 1730
1680 poke tt,239 : poke sc,16 : poke sr,tc
1690 for n = 1 to 50
1700 read nn : poke tt,nn
1710 for z = 1 to 50 : next
1720 nextn : restore : poke sc,0 : return
1730 poke sc,0
1740 tc = 15 : gosub 1680 : gosub 1770
1750 tc = 51 : gosub 1680 : gosub 1770
1760 tc = 85 : gosub 1680 : gosub 1770 : return
1770 for n = 1 to 100 : next : return
1780 data 239, 225, 213, 201, 190, 179, 169
1790 data 159, 150, 142, 134, 127, 119, 113
1800 data 106, 100,  95,  89,  84,  80,  75
1810 data  71,  67,  63,  60,  60,  63,  67
1820 data  71,  75,  80,  84,  89,  95, 100
1830 data 106, 113, 119, 127, 134, 142, 150
1840 data 159, 169, 179, 190, 201, 213, 225, 239
1850 print " *** siren *** "
1860 poke sc,16 : poke sr,15
1870 for n = 1 to 4 : for nn = 250 to 80 step-2 : poke tt,nn
     : next
1880 for nn = 80 to 250 step 2 : poke tt,nn : next : next
1890 poke sc,0
1900 rem return
1910 print " *** ticking clock *** "
1920 m = 150 : mm = 50 : for n = 1 to 21
1930 poke tt,m + mm : poke sc,16 : poke sr,15
     : poke sc,0
1940 for nn = 1 to 250 : next : mm = -mm : next
1950 poke sc,0
1960 rem return
```

The SID In Review

James W. Whitewood
Campbellville, Ont.

In the past, many articles have been written on the unleashing of the SID's power. Most of them were written for the average programmer and in most cases these are all quite readable and digestible. However there are very few that touch upon the uses of such a micro chip.

What Is The SID?

As most of us know the name SID is an acronym for Sound Interface Device. The SID chip can be considered a microprocessor in itself. However, unlike a conventional microprocessor the SID has been designed to perform one function, sound generation. To "program" the SID chip you simply tell it what note you wish to play, when you want it to play and when you want it to stop. Now doesn't that sound easy? The truth of the matter is that, as I'm sure most of you who have played with the SID chip have discovered, there are several bits and pieces that you have to define before you can tell the SID to start playing a note.

The SID chip for all the good press it has received, does leave something to be desired. Consider, for instance, a piano. Let's say we play middle C. When you release the note (ie. lift your finger) you will continue to hear the note even if it is only for a brief time. Now consider the same note with the sustaining pedal pressed. The note will now continue to play for up to 15 seconds after the key has been released. When you release a SID voice (tell it to stop playing) it goes into the release cycle, that is, it starts to decrease the volume until it is down to zero. So far this sounds good, so where is the problem? Well, on a piano if you play a C with the sustain pedal down and then you lift your finger and play an E you will hear the two notes playing together. As a programmer you have no way of knowing when the release cycle has been finished. The temptation therefore is for you to use the same voice for the second note as you did for the first, but when you setup the second note and turn it on, the volume of the first note may not have reached zero. The result is that the first note is cheated out of a proper completion. The simulation of the piano suffers as a result. This premature truncation of the note is not a noticeable problem when a short release time has been programmed into the SID and in some cases it may even be advantageous. You can get around the problem however by setting up a count down table when you release the SID and then waiting for the count down to reach zero before giving the voice a new note to play.

The SID chip also has the undesirable feature of being almost entirely write–only. This means when you tell it something, you can't, at a later time, ask it what it was. In the case of turning the SID voice on or off this is very annoying. As most of you are aware, to turn a SID voice on, you have to set bit 0 in either locations 54276, 54283 or 54290. This may sound straight forward enough, but in order to do this you have to know what the rest of the byte looked like so that you don't change the waveform etc. Again you can get around this problem by keeping a copy of the SID's parameters elsewhere in memory. This solution, in my mind, is awkward and in programs where speed is important this could lead to problems.

The SID chip also has some beautiful filter capabilities. You may ask it NOT to pass certain notes to the outside world. This is useful for when you want to remove the harsh sounds caused by harmonic overtones. This has the effect of mellowing the sound out, giving it a more pleasant sound. A type of vibrato may be done by setting the filter up in its band pass mode and then increasing and decreasing the cutoff frequency of the filter. The interesting thing about the SID chip is that when you alter the filter type you get quite an audible click. In most cases this does not effect anything since filters are usually picked before a piece commences.

I have discovered only one last difficulty while using the SID chip. While the release cycle is supposed to decrease the volume to zero, the chip never really seems to get there. It seems that after a voice has been released the note plays forever at a volume of about .5 on a scale of 0 to 15. This is most annoying when one is demonstrating music to a large audience since there will always be some background noise. My sources at Commodore tell me that they are contemplating altering capacitors C10 and C11 from 2200pf to 1000pf. I do not know whether Commodore found that by changing the capacitors the problem of residual noise goes away or whether they were just speculating that it would.

On the whole, however, I am extremely pleased that Commodore came out with the SID chip. The power of musical synthesis on one chip combined with its relative ease of programming will allow many more young musicians to express themselves.

Sound Maestro

Darren Spruyt
Gravenhurst, Ont.

Sound Maestro is a sound utility for the Commodore–64. It aids in the production of music and/or sound on the Commodore–64. It is also quite easy to use. It eliminates the need to calculate the product of lo nybble + hi nybble x 16 for the envelope paramters. All you need supply is the actual value from 1 to 15.

After the program has been LOADed and RUN, it is ready to use. The format to produce the note is:

SYS49152,VOICE,LO,HI,ATT,DEC,SUS,REL,DUR,WAVE

Where:
VOICE is the voice number (1–3)
LO is the low value of the note to be played
HI is the high value of the note to be played
ATT is the attack rate (0–15)
DEC is the decay rate (0–15)
SUS is the sustain volume (0–15)
REL is the release rate (0–15)

For note values and envelope rates, see The Transactor Vol. 4, #5, p33. . . The Reference Issue.

The DURATION is the time in jiffies (1/60 second) that the note is to pause on the SUSTAIN section of the ENVELOPE. Finally, WAVE is the waveform value for the note to be played ie. 32 = Sawtooth, 16 = Triangle, 64 = Pulse, 128 = Noise.

If using Pulse, set up the pulse setting before–hand. The master volume also needs to be turned on with:

poke 54272 + 24, mv :rem master volume (0–15)

Listing 1 is the machine language loader for Sound Maestro. Listing 2 is a short program using Sound Maestro. LOAD and RUN this after having RUN Sound Maestro.

Editor's Note

Even Darren's own song demo uses only 3 of Sound Maestro's capabilities. Although a simple example is probably best, Sound Maestro paves the way for some pretty advanced SID workouts. With this utility, those frustrating write-only registers in the SID are somewhat more tolerable. Lastly, C64–Link users will have to use the Link Relocator program before running the Sound Maestro loader. If you don't have the Relocator, contact Richvale Telecomm for a copy.

```
10 rem sound maestro–64 v1.00 84/02/06
20 rem darren spruyt
30 rem box 1226 gravenhurst
40 rem ontario, p0c 1g0
100 print:print " Sq sound maestro–64 "
110 print:print " darren spruyt "
120 print:print " setting up. . ..
130 for k = 0 to 427 : read a
150 poke 49152 + k,a : ch = ch + a
160 poke 1024,a : poke 55296,a
170 next
180 if ch<> 48444 then print " checksum error " : end
200 print " Sq instructions for use: "
210 print " q sys 49152,v,lo,hi,att,dec,sus,rel,p,wav "
220 print " where:
230 print " v  = voice number (1–3 inclusive)
240 print " lo  = low  value of note
250 print " hi  = high value of note
260 print " att = attack value (0–15)
270 print " dec = decay value (0–15)
280 print " sus = sustain level (0–15)
290 print " rel = release value (0–15)
300 print " dur = pause (# of jiffies at sustain volume)
310 print " wav = waveform type (16 = tri, 32 = saw,
    64 = pulse, 128 = noise)
320 print " note: "
330 print " pulse, if used, the width must be set up
    before–hand. "
340 print " master volume must also be turned on. "
350 print " if i/o troubled, run–stop/restore.
360 print " note and att/dec/sus/rel values are on  p33,
    in transactor vol4, #5
1000 data  32, 254, 192, 168, 208,   3,  76,   8
1010 data 175, 201,   4, 176, 249, 202, 142,  94
1020 data 193, 142,  95, 193, 138,  10,  10,  10
```

```
1030 data  56, 237,  94, 193, 141,  94, 193,  24
1040 data 105,   6, 168, 169,   0, 153,   0, 212
1050 data 136, 136, 169,   0, 153,   0, 212, 160
1060 data   2, 162, 255, 202, 208, 253, 136, 208
1070 data 248,  32,  13, 193, 172,  94, 193, 153
1080 data   0, 212, 238,  94, 193,  32,  13, 193
1090 data 172,  94, 193, 153,   0, 212, 173,  94
1100 data 193,  24, 105,   4, 141,  94, 193,  32
1110 data 254, 192,  10,  10,  10,  10, 141,  96
1120 data 193,  32, 215, 192,  32, 254, 192,  13
1130 data  96, 193, 172,  94, 193, 153,   0, 212
1140 data  32, 231, 192, 238,  94, 193,  32, 254
1150 data 192,  10,  10,  10,  10, 141,  96, 193
1160 data  32, 254, 192,  13,  96, 193, 172,  94
1170 data 193, 153,   0, 212,  24,  32, 115,   0
1180 data  32, 158, 173,  32, 247, 183,  24, 172
1190 data  95, 193, 165,  20, 121,  85, 193, 153
1200 data  85, 193, 165,  21, 121,  88, 193, 153
1210 data  88, 193, 173,  94, 193,  56, 233,   2
1220 data 141,  94, 193,  32,  13, 193,  41, 254
1230 data 172,  95, 193, 153,  97, 193, 172,  95
1240 data 193, 190,  97, 193, 232, 138, 172,  94
1250 data 193, 153,   0, 212, 172,  95, 193, 169
1260 data   1, 153,  91, 193,  76, 103, 193, 172
1270 data  95, 193, 189,  21, 193, 153,  85, 193
1280 data 189,  37, 193, 153,  88, 193,  96, 172
1290 data  95, 193,  24, 189,  53, 193, 121,  85
1300 data 193, 153,  85, 193, 189,  69, 193, 121
1310 data  88, 193, 153,  88, 193,  96,  32, 253
1320 data 174,  32, 158, 183, 138, 201,  16, 176
1330 data   1,  96,  76,   8, 175,  32, 253, 174
1340 data  32, 158, 183, 138,  96,   1,   1,   1
1350 data   2,   3,   4,   5,   5,   6,  15,  30
1360 data  48,  60, 180,  44, 224,   0,   0,   0
1370 data   0,   0,   0,   0,   0,   0,   0,   0
1380 data   0,   0,   0,   1,   1,   1,   2,   3
1390 data   5,   7,  11,  13,  15,  18,  45,  90
1400 data 144, 180,  28, 132, 160,   0,   0,   0
1410 data   0,   0,   0,   0,   0,   0,   0,   0
1420 data   0,   0,   2,   3,   5, 255, 255, 255
1430 data 255, 255, 255,   0,   0,   0,  18,   2
1440 data 112,  32,  32,  32,   4,  11,  18, 120
1450 data 169, 117, 141,  20,   3, 169, 193, 141
1460 data  21,   3,  88,  96,  96, 160,   0, 185
1470 data  91, 193, 240,  39, 190,  85, 193, 202
1480 data 138, 153,  85, 193, 201, 255, 208,  27
1490 data 190,  88, 193, 202, 138, 153,  88, 193
1500 data 201, 255, 208,  15, 185, 100, 193, 170
1510 data 185,  97, 193, 157,   0, 212, 169,   0
1520 data 153,  91, 193, 200, 192,   3, 208, 207
1530 data  76,  49, 234, 234
```

```
10 rem quarter master's store
20 rem using darren spruyt's
30 rem sound maestro v1.00
100 poke 54272 + 24,10 : rem volume 10
110 zz = 900 : xx = -1 : rem set xx = 0 for fun
120 read lo, hi, dur
130 if lo<0 then stop
140 sys 49152, 1, lo, hi, 0, 0, 9, 0, 900*dur, 32
150 if xx then 180
160 sys 49152, 2, lo + 5,  hi,0,0,9,0,900*dur,32
170 sys 49152, 3, lo + 10,hi,0,0,9,0,900*dur,32
180 for j = 0 to zz*dur : next : goto 130
1000 rem data for the notes
1010 rem measure #1
1020 data 97,8,.25
1030 rem measure #2
1040 data  48,  11,  .25
1050 data  48,  11,  .25
1060 data  48,  11,  .25
1070 data 143,  10,  .25
1080 rem measure #3
1090 data 104, 9, .25
1100 data 104, 9, .25
1110 data 104, 9, .25
1120 data 104, 9, .25
1130 rem measure #4
1140 data 143,  12, .25
1150 data  24,  14, .25
1160 data 143,  12, .25
1170 data  48,  11, .25
1180 rem measure #5
1190 data 143, 10, .25
1200 data 143, 12, .25
1210 data 195, 16, .50
1220 data  97,  8, .25
1230 rem measure #6
1240 data  48, 11, .25
1250 data  24, 14, .25
1260 data  48, 11, .25
1270 rem measure #7
1280 data 104,   9, .25
1290 data 210,  15, .25
1300 data 143,  12, .25
1310 rem measure #8
1320 data 210,  15, .25
1330 data 143,  10, .25
1340 rem measure #9
1350 data  48,  11, .75
1360 data  97,   8, .375
1370 data  48,  11, .0625
1380 rem measure #10
1390 data  24,  14, .25
1400 data   0,   0, .25
1410 data  24,  14, .25
1420 data   0,   0, .25
1430 rem measure #11
1440 data  24,  14, .375
1450 data 143,  12, .0625
1460 data  48,  11, .375
1470 data 104,   9, .0625
1480 data  97,   8, .25
1490 data 143,  10, .375
1500 data  48,  11, .0625
1510 rem measure #12
1520 data 143,  12, .75
1530 data  48,  11, .375
1540 data 143,  12, .0625
1550 rem measure #13
1560 data  24,  14, .75
1570 data  97,   8, .375
1580 data  48,  11, .0625
1590 rem measure #14
1600 data  24,  14, .25
1610 data   0,   0, .25
1620 data  24,  14, .25
1630 data   0,   0, .25
1640 rem measure #15
1650 data  24,  14, .375
1660 data 143,  12, .0625
1670 data  48,  11, .375
1680 data 104,   9, .0625
1690 data  97,   8, .25
1700 data 143,  10, .375
1710 data  48,  11, .0625
1720 rem measure #16
1730 data 143,  12, .3
1740 data 210,  15, .3
1750 data  24,  14, .3
1760 data 143,  12, .3
1770 rem measure #17
1780 data  24,  14,  1
1790 data   0,   0,  1
1800 data  -1,  -1, -1
```

Sprite Rotate

Chris Zamara
Downsview, Ont.

Rotational Animation The Easy Way!

Sprites in the C64 are excellent for animation. By defining a number of different sprite shapes in memory, then storing the pointers to those shapes sequentially into the VIC–II video chip, a shape on the screen can appear to undergo a metamorphosis. One of the common applications of this technique is when a shape must appear to rotate, either clockwise or counter–clockwise. For example: in an asteroids–type video game, the player's spaceship must rotate to the position in which it is supposed to travel.

The best way to accomplish this is to define as many sprites as there are rotational positions of the ship (32 is a good number), where each of the 32 sprite definitions looks like the ship pointed in one of its 32 positions. The video game program then merely changes the sprite shape pointer in the VIC chip, and the ship appears to rotate. The code for the game becomes easy, but the hard part is defining all of those sprites (32 of them!), and defining them accurately, so that the ship rotates smoothly and naturally.

It is for exactly the above reason that I wrote "Sprite Rotate". Sprite Rotate takes a single sprite, and rotates it clockwise to create as many sprites as you need, completing a full 360 degrees of rotation.

A single sprite is defined (using Commodore's sprite editor, perhaps) in any "sprite page". After running Sprite Rotate, you will first be asked the centre of rotation of the sprite. The centre of a 24 x 21 sprite grid is (12,11), and that is the usual reply for the first question. You will then be asked the sprite page of the shape. Each sprite page is 64 bytes long, and your sprite definition must begin on a 64 byte boundary. The address of the start of the shape divided by 64 will give the sprite page. All of the new (rotated) shapes produced by sprite rotate will be stored in subsequent sprite pages, and the original sprite will be the first in the animation sequence.

The next question to answer is the number of positions to rotate the sprite. Our above spaceship example would require 32. Make sure you have space in memory above the page of the original sprite. You may request as many as you want (more positions give smoother movement), but be forewarned: the program takes a LONG time to execute and each rotation means 64 more bytes of memory get used up.

The last question is whether you want to SAVE the sprite data. If you answer 'Y', you will be prompted for a filename, and a program file will be saved on disk, containing the original sprite as well as all of the generated sprites. This file can be LOADed with:

```
LOAD "filename", 8, 1
```

Don't forget the ",1" for non–relocating load.

Once the above information has been supplied, Sprite Rotate will start chugging away, rotating each shape point by point. You can see the current sprite being worked on, and the next sprite in the sequence being created on the screen. The process is quite slow, but the program can be optimized, as it is now written to be easily read and understood, not to go fast. After the rotation process is complete, the sprite can be seen in action. Press the F1 and F7 keys as indicated to see the sprite rotate clockwise or counter clockwise. At this point, pressing the space bar ends the program.

If you have occasion to use this program just once, it is worth the effort of typing it in. I only used it once, but even all the effort taken to write it amounted to less time than it would take to accurately draw 40 sprites using the sprite editor.

```
100 rem ************************************
105 rem " *rotate sprite program - Chris Zamara, 1983*
110 rem " *        Creates N sprite shape tables,    *
115 rem " *          rotating a sprite clockwise.    *
120 rem ************************************
125 sc$=chr$(147) : cd$=chr$(17)
130 print sc$" enter centre of rotation (x,y) ";
135 input cx,cy
140 input " sprite page of shape " ;begin
145 input " number of positions for rotation " ;n
150 s=0: input " save sprite data (y/n) " ;y$
155 if left$(y$,1)<>" y " then 175
160 input " filename " ;f$
165 s=-1
170 :
175 g=13*4096
180 poke g+21,3
185 poke 2040,200 : poke 2041,201
190 poke g+39,1   : poke g+40,1
195 poke g+0,90   : poke g+2,180
200 poke g+16,0
205 poke g+1,60   : poke g+3,60
210 poke g+29,0
215 poke g+23,0
220 poke g+28,0
225 print sc$;
230 for i = 0 to 7 : e(i)=2↑(7-i) : next i
235 :
240 baddr = begin*64
245 print " hold on. clearing sprite area "
250 for i = (begin+1)*64 to (begin+n)*64
255 poke i,0 : next i
260 print sc$;
265 rtate = -2* 3.1415926 / n: angle = rtate
    : rem * use pi symbol
270 page = begin
275 :
280 for i = 1 to n-1
285 addr = baddr
290 :
295 for y0 = 0 to 20
300 x0 = 0
305 for col = 0 to 2
310 for bit = 0 to 7
315 :
320 if (peek(addr) and e(bit)) = 0 then 355
325 :
330 x = cx-x0 : y = cy-y0
335 x2 = x*cos(angle)-y*sin(angle)
340 y2 = x*sin(angle)+y*cos(angle)
345 gosub 505
350 :
355 x0 = x0 + 1
360 next bit
365 addr = addr + 1
370 next col
375 next y0
380 :
385 angle = angle + rtate
390 page = page + 1
395 poke 2040,page : poke 2041,page + 1
400 print i
405 next i
410 :
415 if s=0 then 480
420 open1,8,12, " @0: " +f$+ " ,p,w "
425 sh = int(baddr/256)
430 sl = baddr-sh*256
435 print#1,chr$(sl)chr$(sh);
440 rem * start address for load *
445 for i = baddr to (begin+n)*64
450 print#1,chr$(peek(i));
455 next i
460 close1
465 print:print " ok, file is saved. use ,8,1 after load
    to load it starting " ;
470 print " at the same page the first sprite came from.
475 :
480 gosub 570
485 end
490 :
495 rem ****** end of mainline *******
500 :
505 rem * point plot subroutine
510 x2 = int(x2+cx+0.5)
515 y2 = int(cy-y2+0.5)
520 if x2>23 or y2>20 or x2<0 or y2<0 then return
525 :
530 cl = int(x2/8)
535 bt = e(x2 and 7)
540 a2 = (page+1)*64+cl+y2*3
545 if a2> 64*256 then print " warning: out of
    bound address " : return
550 poke a2,peek(a2) or bt
555 return
560 :
565 rem * view rotating sprites
570 print sc$ cd$ cd$ cd$ cd$ cd$;
575 print " press f1 or f7 to rotate shape "
580 print cd$ " press space to end. "
585 a = begin : en = begin + n-1
590 poke 2040,a
595 k=peek(197)
600 if k=4 then a=a-1
605 if k=3 then a=a+1
610 if k=60 then return
615 if a<begin then a=en
620 if a>en then a=begin
625 for d=1 to 50/n: next d
630 goto590
```

QUASIMOB

Chris Zamara
Downsview, Ontario

Break The 8 Sprite Barrier!

One of the great things about the VIC–II video chip in the Commodore 64 is its use of sprites, or moveable object blocks (MOBs). Using sprites in a game program greatly simplifies the task that the software normally has to perform, and generally results in a faster moving, higher quality game.

What could be better than sprites? More sprites! The VIC video chip allows up to 8 sprites at a time to be defined, which means that in normal use, a maximum of 8 sprites may appear on the screen at one time. (any number of sprite definitions may exist at any one time, but only 8 of these may be selected). But wait, some of you are saying, Commodore claims that up to 8 sprites may be defined per line. How is this possible? Well, first let's look at how it is possible to display more than 8 sprites on the screen, and the limitations imposed by this technique.

How to Display More Sprites

First of all, how can more than 8 sprites be seen on the screen at the same time? Well, the trick is to use the 64's powerful **raster interrupt** feature. The VIC chip can be programmed to interrupt the execution of instructions at any given raster scan line (sprite vertical or Y position), and have the CPU execute a user routine at that point before continuing with what it was doing. After the raster beam draws the first set of 8 sprites (assuming all 8 are used), the user interrupt routine simply sets up new sprite definitions (shape pointer, coordinates, colour, etc.), and the raster beam draws them. The previous definitions remain on the screen while the new ones go on, but the original set of sprites must be redefined before the picture tube's electron beam gets back to the top of the screen again so the first 8 sprites can be re–drawn. There will be no visible flicker, since the beam is always tracing over valid sprite definitions.

Limitations of the Raster Technique

This basic technique works fine, except there are certain problems: what would happen if a sprite moved down so that part of it was below an interrupt line, and part was above? Well, if the sprite was redefined below the line, it would be "cut off" at the line, and the bottom half would probably appear in some strange place on the screen. If a program tried to move a set of sprites freely about the screen, all kinds of silly things could happen. Thus, using the raster interrupt technique clearly limits a set of sprites to a definite field.

Sprite Fields

The question is, how many scan lines high should these fields be? The smallest field would give the maximum number of sprites on the screen, since the screen could be divided into the most fields, and each field could contain a maximum of 8 sprites.

The smallest field size is determined by the height of a sprite. This height is 21 for a regular sized sprite, or 42 for a vertically expanded sprite. If we make a field exactly as high as a sprite, however, the sprite will not be able to move up or down without moving into another field. If the same sprite is redefined in the field above or below, and we must assume the worst case, part of the sprite will get "cut off". This is the problem with the "8 sprites per line" claim. A bigger sprite field would give the sprite more freedom of movement, but it would of course limit the maximum number of sprites that could appear on the screen. It looks like a trade–off will have to be made.

Enter "QuasiMOB"

Since each application of sprites has unique requirements, I have written "QuasiMOB", a utility which allows any number of different sprite configurations.

The number of sprite fields on the screen, and the start raster line of each field is variable. The other variable is the number of "quasi sprites". A quasi sprite is used just like a normal sprite, but instead of storing the sprite's coordinates, colour, shape, etc. directly into the VIC chip as usual, these parameters are stored in a special 64 byte area of memory. QuasiMOB stores the parameters from the appropriate quasi sprite into the VIC chip at the correct scan line, making one sprite into as many sprites as there are fields.

An Example: Using 12 sprites

For example: suppose we need 12 sprites on the screen for a particular application, and four of these sprites must have complete freedom of movement on the screen (i.e. the software can never guarantee where one of these four sprites will be). These 4 sprites must be "real" sprites, which are defined in the VIC chip as usual. Since we need an additional 8 sprites, we can allocate 4 quasi sprites, and split the screen into 2 fields. The boundary between the two fields will be in the middle of the screen, at scan line 135. This gives us 4 sprites which must stay in the first field, or top half of the screen, and 4 sprites in the bottom half. The general rule for sprite movement is that a sprite must be 14 or more scan lines (Y units) away from a border. So in this case, the top sprites could have Y values from 14 to 121 (maximum is 135 minus 14), and the bottom sprites could have Y values from 149 to 241 (minimum is 135 plus 14). This 12 sprite scheme works quite well for many games, and 12 sprites at one time is often all you need.

To implement the above scheme, QuasiMOB needs to know the following:

- the number of quasi sprites to use: 4.
- the number of fields: 2.
- the start raster lines of each field, which is 0 for field 1 (top of screen), and 135 for field 2 (approximately the middle of the screen).

The above parameters could be set up from BASIC in the following way:

```
POKE 49155,4
POKE 49156,2
POKE 49157,0     : rem field 1 = 0, top line of screen
POKE 49158,135 : rem field 2 = 135, approx. half way
                                    down.
```

This is the way that QuasiMOB is set up in the listing in this article. The 4 quasi sprites, sprites 0–3 in the VIC chip, will be defined and redefined by QuasiMOB during the interrupts, and sprites 4–7 will be available for normal use. Thus, to use one of the 4 quasi sprites in the top half of the screen, change the "sprite set 1" area of memory, and to change any of the sprites in the bottom half of the screen, use the "sprite set 2" area.

(Remember, this is only one example. You could select 8 quasi sprites and 3 fields for a total of 24, but this would allow for no "real" sprites and the quasi sprites would be limited to smaller fields. Additional field borders (12 have been allowed for) go into bytes 49159 through 49168. Refer to the listing to see how the parameters in the quasi sprite memory areas are set up. Near the end of the listing, all the parameters for field 1 are defined and explained.

Running QuasiMOB

Once this information has been supplied, QuasiMOB is enabled with:

SYS 49152

After this initialization, the system returns to BASIC, and the program runs during the interrupts, invisible to the user. If the sprite parameters are set up as in the listing, 8 sprites should appear on the screen (since no shapes have been defined, the sprites may look like a block of garbage, but don't worry about it). Four sprites will be in the top half of the screen, and 4 in the bottom half. Remember that only the first 4 sprites in the VIC chip are being used, so you can still put another 4 sprites on the screen. This gives us the illusion of 12 sprites: the 2 sets of 4 "quasi" sprites (0–3 x 2), and the 4 "real" sprites (4–7).

A BASIC or machine language program can modify the contents of the quasi sprite memory area, and move around or change the shapes of the quasi sprites on the screen. For example, in the above setup using 4 quasi sprites and 2 fields, to change the x coordinate of quasi sprite 1 in the top half of the screen (field 1), you could use POKE 49451,x from BASIC (where 'x' is the x coordinate for the sprite). This changes the first byte in the quasi sprite memory area, which is the x coordinate of quasi sprite 1, field 1. The field 2 parameters start 64 bytes later, at 49515, and the parameters for subsequent fields (if there are more than 2) continue at 64 byte intervals. Once again, refer to the assembler listing to see which bytes to change for each sprite parameter.

Note that after QuasiMOB has been initialized, normal control of the system is restored in most respects. SAVEing

or LOADing to or from disk or tape will not work properly while QuasiMOB is running, so RESTORE before such an operation, and re-enable QuasiMOB (with SYS 49152) afterwards.

Each quasi sprite area is 64 bytes long, and contains all the information for 8 sprites. It is set up like this:

X,Y table (first 16 bytes)
X high bits (17th byte)
X expand (18th byte)
Y expand (19th byte)
sprite enable (20th byte)
colour (21st to 28th bytes)
shape (29th to 36th bytes)

Some of the other features, such as sprite multi colour mode are not supported, but such features can be easily added if needed (only the first 36 bytes in each 64 byte area are now being used). Remember that each additional feature slows down program execution time slightly, since the CPU has to do more at every interrupt. If less than 8 quasi sprites are being used, the first parameters in each category are used. Thus, in the above example using 4 quasi sprites, only the first 4 X,Y coordinates, the first 4 colours, etc. in each quasi sprite area need be defined. The rest will be ignored. The single bit parameters (x high bits, sprite enable, and X & Y expand) work in the same way as in the VIC chip: the least significant bit corresponds to sprite 0, bit 1 corresponds to sprite 1, etc.

Conclusion

QuasiMOB gives an easy way to add extra sprites to programs that need them. The program is intended as an example on how to use more than 8 sprites, and so is quite simple. Readers who are not too familiar with the use of raster interrupts can learn more about them by looking at the code and reading the comments in the program. You should be able to easily add any extra features that you need, and use the program as a base for more complex sprite control programs (maybe to allow sprite movement from one field to another through some priority sorting technique). Using more than 8 sprites on the Commodore 64 is a good example of how software can overcome hardware limitations.

```
1000 rem quasimob loader
1010 for j=49152 to 49550 : read x
1020 poke j,x : ch=ch+x : next
1030 if ch<> 35713 then print " checksum error "
1040 rem
1050 data  76,  39, 192,   4,   2,   0, 135,   0
1060 data   0,   0,   0,   0,   0,   0,   0,   0
1070 data   0,   0,   0,   0,   0, 255, 254, 252
1080 data 248, 240, 224, 192, 128,   0,   0,   1
1090 data   3,   7,  15,  31,  63, 127, 255, 169
1100 data 127, 141,  13, 220, 169,   1, 141,  26
1110 data 208, 120, 173,  17, 208,  41, 127, 141
1120 data  17, 208, 169,   0, 141,  18, 208, 141
1130 data  17, 192, 169,  78, 141,  20,   3, 169
1140 data 192, 141,  21,   3,  88,  96, 173,  25
1150 data 208,  41,   1, 208,   3,  76, 134, 192
1160 data 169,   1, 141,  25, 208,  32, 140, 192
1170 data 238,  17, 192, 174,  17, 192, 236,   4
1180 data 192, 208,   5, 162,   0, 142,  17, 192
1190 data 173,  17, 208,  41, 127, 141,  17, 208
1200 data 189,   5, 192, 141,  18, 208, 173,  17
1210 data 192, 208,   3,  76,  49, 234, 104, 168
1220 data 104, 170, 104,  64, 172,   3, 192, 173
1230 data  21, 208,  57,  21, 192, 141,  21, 208
1240 data 173,   3, 192,  10, 141,  20, 192, 162
1250 data   0, 173,  17, 192,  10,  10,  10,  10
1260 data  10,  10, 141,  18, 192, 168, 185,  43
1270 data 193, 157,   0, 208, 200, 232, 236,  20
1280 data 192, 208, 243, 174,  18, 192, 172,   3
1290 data 192, 173,  16, 208,  57,  21, 192, 141
1300 data  19, 192, 189,  59, 193,  57,  30, 192
1310 data  13,  19, 192, 141,  16, 208, 173,  23
1320 data 208,  57,  21, 192, 141,  19, 192, 189
1330 data  61, 193,  57,  30, 192,  13,  19, 192
1340 data 141,  23, 208, 173,  29, 208,  57,  21
1350 data 192, 141,  19, 192, 189,  60, 193,  57
1360 data  30, 192,  13,  19, 192, 141,  29, 208
1370 data 162,   0, 172,  18, 192, 185,  63, 193
1380 data 157,  39, 208, 185,  71, 193, 157, 248
1390 data   7, 200, 232, 236,   3, 192, 208, 237
1400 data 174,  18, 192, 172,   3, 192, 189,  62
1410 data 193,  57,  30, 192,  13,  21, 208, 141
1420 data  21, 208,  96,  10,  80, 150,  60,  80
1430 data  90,  30, 110,   0,   0,   0,   0,   0
1440 data   0,   0,   0,   1,   2,   2,  15,   1
1450 data   2,   3,   4,   0,   0,   0,   0, 128
1460 data 129, 130, 131,   0,   0,   0,   0,   0
1470 data   0,   0,   0,   0,   0,   0,   0,   0
1480 data   0,   0,   0,   0,   0,   0,   0,   0
1490 data   0,   0,   0,   0,   0,   0,   0,   0
1500 data   0,   0,   0, 190, 150,  10, 200, 180
1510 data 170,  70, 160,   0,   0,   0,   0,   0
1520 data   0,   0,   0,   2,   8,   8,  15,   5
1530 data   6,   7,   8,   0,   0,   0,   0,   0
1540 data  33,  55,  55,   0,   0,   0,   0
```

QuasiMOB Source Listing

```
c000                    *= $c000
c000 4c 27 c0           jmp start
c003 04        sprtno   .byte4          ;# of quasi sprites
c004 02        fieldno  .byte2          ;# of fields
c005 00 87 00 fields    .byte0,135,0,0,0,0,0,0,0,0,0,0
                        ;start line of each field
c011           vic      = $d000         ;vic chip start address
c011 00        intcount .byte0          ;interrupt count
c012 00        pntr     .byte0
c013 00        temp     .byte0
c014 00        sprtno2  .byte0
c015 ff fe fc  mask1    .byte255,254,252,248,240,224,192,128,0
c01e 00 01 03  mask2    .byte0,1,3,7,15,31,63,127,255
                        ;**initialization code starts here**
c027           start    = *
c027 a9 7f              lda #$7f
c029 8d 0d dc           sta $dc0d       ; disable timer as interrupt source
c02c a9 01              lda #1
c02e 8d 1a d0           sta vic + 26    ; enable raster as interrupt source
c031 78                 sei
c032 ad 11 d0           lda vic + 17    ;raster compare
c035 29 7f              and #%01111111 ;register, high bit
c037 8d 11 d0           sta vic + 17
c03a a9 00              lda #0          ;set raster reg. to
c03c 8d 12 d0           sta vic + 18    ;top of screen
c03f 8d 11 c0           sta intcount    ;initialize counter
c042 a9 4e              lda #<intrtn
c044 8d 14 03           sta $0314       ; set new interrupt vector
c047 a9 c0              lda #>intrtn
c049 8d 15 03           sta $0315
c04c 58                 cli
c04d 60                 rts
c04e           intrtn   = *
                        ;*interrupt routine - performed when beam is
                        ;*at one of the pre-set field boundaries.
c04e ad 19 d0           lda vic + 25
                        ; only process interrupts caused by
                        ; raster beam compare register
c051 29 01              and #1
c053 d0 03              bne myirq       ; raster interrupt
c055 4c 86 c0           jmp getout
                        ;ignore this interrupt if not caused by raster
c058           myirq    = *
                        ;clear interrupt flag
c058 a9 01              lda #1
c05a 8d 19 d0           sta vic + 25
                        ;*raster beam is at boundary
c05d           fieldint = *
                        ;select appropriate quasi sprites
c05d 20 8c c0           jsr sprtset
c060 ee 11 c0           inc intcount
c063 ae 11 c0           ldx intcount
c066 ec 04 c0           cpx fieldno
c069 d0 05              bne noend
                        ;reset counter if at last field
c06b a2 00              ldx #0
c06d 8e 11 c0           stx intcount
c070           noend    = *
c070 ad 11 d0           lda vic + 17    ;set msb of raster
c073 29 7f              and #%01111111 ;register to 0
c075 8d 11 d0           sta vic + 17
c078 bd 05 c0           lda fields,x    ;set raster to interrupt
c07b 8d 12 d0           sta vic + 18    ;at next field
c07e ad 11 c0           lda intcount
c081 d0 03              bne nosysirq
                        ; perform system irq routine after last field
                        ; (system irq routine should be completed by
                        ; the time the beam is at the top of the screen)
c083 4c 31 ea           jmp $ea31
c086           nosysirq = *
c086           getout   = *            ; exit from interrupt
c086 68                 pla
c087 a8                 tay             ;restore y register,
c088 68                 pla
c089 aa                 tax             ;x register,
c08a 68                 pla             ;and accumulator, then
c08b 40                 rti             ;return from interrupt
                        ;**the quasi sprite setup routine follows.**
                        ;'sprtset' uses the quasi sprite table indicated
                        ;by 'intcount', and uses the values in that table
                        ;as parameters for real sprites.
c08c           sprtset  = *
c08c ac 03 c0           ldy sprtno
c08f ad 15 d0           lda vic + 21
c092 39 15 c0           and mask1,y
c095 8d 15 d0           sta vic + 21
                        ;disable sprites to be used
                        ;x y table (8 bytes)
c098 ad 03 c0           lda sprtno
c09b 0a                 asl a
c09c 8d 14 c0           sta sprtno2
                        ;number of sprites per field
c09f a2 00              ldx #0          ;pointer to vic-ii address
c0a1 ad 11 c0           lda intcount    ;pointer to table
c0a4 0a                 asl a
c0a5 0a                 asl a
c0a6 0a                 asl a
c0a7 0a                 asl a
c0a8 0a                 asl a
c0a9 0a                 asl a
c0aa 8d 12 c0           sta pntr
c0ad a8                 tay  ;y is count*64
                        ;'pntr' points to quasi sprite
c0ae b9 2b c1 stp1      lda xy,y
c0b1 9d 00 d0           sta vic,x
c0b4 c8                 iny
c0b5 e8                 inx
c0b6 ec 14 c0           cpx sprtno2
c0b9 d0 f3              bne stp1
c0bb ae 12 c0           ldx pntr
c0be ac 03 c0           ldy sprtno
                        ; for all the following bit-switch parameters, the
                        ; low bits from the quasi sprite parameter, and the
                        ; high bits from the value in the vic chip are stored
                        ; into the vic chip. the number of the low and high
                        ; bits used depends on the number of quasi sprites
                        ;x most significant bits
c0c1 ad 10 d0           lda vic + 16
c0c4 39 15 c0           and mask1,y
c0c7 8d 13 c0           sta temp
c0ca bd 3b c1           lda xhi,x
c0cd 39 1e c0           and mask2,y
c0d0 0d 13 c0           ora temp
c0d3 8d 10 d0           sta vic + 16
                        ;y expand
c0d6 ad 17 d0           lda vic + 23
c0d9 39 15 c0           and mask1,y
c0dc 8d 13 c0           sta temp
c0df bd 3d c1           lda expy,x
c0e2 39 1e c0           and mask2,y
c0e5 0d 13 c0           ora temp
c0e8 8d 17 d0           sta vic + 23
                        ;x expand
c0eb ad 1d d0           lda vic + 29
c0ee 39 15 c0           and mask1,y
c0f1 8d 13 c0           sta temp
c0f4 bd 3c c1           lda expx,x
c0f7 39 1e c0           and mask2,y
c0fa 0d 13 c0           ora temp
c0fd 8d 1d d0           sta vic + 29
                        ;set up colours
c100 a2 00              ldx #0
c102 ac 12 c0           ldy pntr
c105 b9 3f c1 colr1     lda colour,y
c108 9d 27 d0           sta vic + 39,x
c10b b9 47 c1           lda shape,y
c10e 9d f8 07           sta 2040,x
c111 c8                 iny
c112 e8                 inx
c113 ec 03 c0           cpx sprtno
c116 d0 ed              bne colr1
                        ;finally, enable proper sprites
c118 ae 12 c0           ldx pntr
c11b ac 03 c0           ldy sprtno
c11e bd 3e c1           lda enbl,x
c121 39 1e c0           and mask2,y
c124 0d 15 d0           ora vic + 21
c127 8d 15 d0           sta vic + 21
c12a 60                 rts
                        ;*memory area for the quasi sprite parameters follows*
                        ;**parameters for first set of 4 " quasi sprites " **
                        ;— x,y table —
c12b 0a 50 96 xy        .byte10,80, 150,60, 80,90, 30,110
c133 00 00 00           .byte0,0, 0,0, 0,0, 0,0
                        ;— x high bits —
c13b 01        xhi      .byte%00000001
                        ;— x expansion bits (1 = expand) —
c13c 02        expx     .byte%00000010
                        ;— y expansion bits —
c13d 02        expy     .byte%00000010
                        ;— sprite enable bits (1 = enable)
c13e 0f        enbl     .byte%00001111
                        ;— colours —
c13f 01 02 03 colour    .byte1,2,3,4,0,0,0,0
                        ;— shape table pointers —
c147 80 81 82 shape     .byte128,129,130,131
c14b 00 00 00           .byte0,0,0,0
                        ;**parameters for quasi sprite set 2**
c16b                    *= xy + 64
c16b           qs2      = *
c16b be 96 0a           .byte190,150, 10,200, 180,170, 70,160
c173 00 00 00           .byte0,0, 0,0, 0,0, 0,0
c17b 02                 .byte%00000010  ;x high bits
c17c 08                 .byte%00001000  ;x expand
c17d 08                 .byte%00001000  ;y expand
c17e 0f                 .byte%00001111  ;sprite enable
c17f 05 06 07           .byte5,6,7,8,0,0,0,0  ;colours
c187 00 21 37           .byte000,033,055,055  ;shapes
c18b 00 00 00           .byte0,0,0,0
c1ab                    *= qs2 + 64
c1ab           qs3      = *
                        ;**parameters for quasi sprite 3 go here
c1eb                    *= qs3 + 64
c1eb           qs4      = *
                        ;**parameters for quasi sprite 4
c22b                    *= qs4 + 64
c22b           qs5      = *
                        ;etc. (define as many as needed)
                        ;the above sprite parameters may set up or changed from
                        ;basic (via a poke), or from a machine language program
                        ; the current settings of quasi sprites 1&2
                        ; are merely for demonstration purposes.
                        ;**********end of source**********
```

Changing The C64's Screen Colours

Bob Drake
Brantford, Ont.

I use a colour monitor with my C64 and I find the usual screen colours are not those I want. Some colours of borders, screens and type do not go together well. Some text colours are virtually impossible to read. When I use a television, black and white or colour, I find the same problem exists with different values than my monitor uses.

Here is a short routine, under 350 bytes, that will allow you to use the function keys to change the border, screen and text colours until they meet your satisfaction. It is easily added to your own program.

Line 102 sets the initial values for the 3 variables. By using a vector (an array with only one row) we don't have to repeat line 118 for each variable. Line 104 sets up the original screen with a black border, black screen and white text. You can set the initial colours by changing line 102. V(1) is the border, V(2) is the screen and V(3) is the character colour. Line 106 sets upper case with the CHR$(142) and prints the basic instructions. Line 110 gets a character from the keyboard. Line 112 converts the key to its ASCII value. The CHR$(0) is concatenated to KEY$ to avoid a problem if no key is pressed. Since the function keys that we want have ASCII values of 133–137, the 132 subtracted drops legal values of K to between 1 and 4. A value of 1, 2, or 3 increases the corresponding V(K) value by one. A check is made to keep K in the right range (0–15). K = 4 causes an exit. Exiting erases all variables with the CLR, freeing the space taken by the array V(K).

```
100 rem adjust c64 screen colour-r.drake 1983
102 v(1)=0:v(2)=0:v(3)=1
104 poke 53280,v(1):poke 53281,v(2):poke646,v(3)
106 print chr$(142) " Sr adjust screen colours "
108 print " f1:border  f3:screen  f5:type  f7:exit
110 get key$
112 k=asc(key$ + chr$(0))-132
114 if k<1 or k>4 then 110
116 if k=4 then 122
118 v(k)=v(k)+1:if v(k)=16 then v(k)=0
120 goto 104
122 clr
```

For those concerned with saving a few bytes here is a packed version, about 100 bytes smaller.

```
1 v(1)=0:v(2)=0:v(3)=1
2 poke 53280,v(1):poke 53281,v(2):poke646,v(3)
3 print " Sr adjust screen colours "
  :print " f1:border  f3:screen  f5:type  f7:exit "
4 get k$:k=asc(k$+chr$(0))-132:if k<1 or k>4 then 4
5 if k=4 then 8
6 v(k)=v(k)+1:if v(k)=16 then v(k)=0
7 goto 2
8 clr
```

Simple Harmonic Motion

PET Graphics Revisited

With the advent of the VIC 20 and Commodore 64, Commodore users world wide have apparently forgotten the inherent graphic capabilities of all the Commodore machines. The purpose of this article is to bring this fact to your attention, and allow you to actually see how terrific PET graphics can be.

Normal screen resolution on the 40 column Commodore computers is 40 characters wide by 25 lines deep, or 1000 characters. On the 8000/9000 machines, this has been increased to 2000 characters. Pretty boring stuff. Well, as numerous authors have pointed out in the past, clever manipulation of the Commodore character set can bring to you a pseudo high resolution display many times greater than normal. For example, in the Commodore character set are 8 "bar" characters that are vertically one pixel apart. They are screen characters 100, 82, 70, 64, 67, 68, 69 and 99 respectively. In the same context, there are also 8 bar characters that are horizontally 1 pixel apart. These are 101, 84, 71, 66, 93, 72, 89 and 103 respectively.

Below is a program written to show just how good your computer can be in the graphics department, even without actual high resolution graphics. And, for an encore, this program could actually be used by many schools and institutions that would like to supply more than a textbook example of assorted wave forms. In using this creation, you are able to plot, a sine, cosine, tangent and logarithmic waveform. With further modifications, any wave form or graphic display could be created. Just let your imagination take over.

At the very end of this article, below the program, are a number of complex mathematical formulas that can be reproduced in the language of Commodore, also known as Commspeak. With these formulas at hand, every wave form imaginable can be reproduced with fair accuracy. You will find, though, that the major stumbling block in your way is the screen width of your computer. The larger the screen, the better the display.

```
100 rem ***************************************
105 rem *          pseudo high res graphics     *
110 rem *          rte - transactor magazine    *
115 rem * cbm 8000 + sp9000 : sw=80:se=34768    *
120 rem * pet 4000 + c64 : sw=40:se=33767      *
125 rem * 4.0 basic        : ss=32768:nc=158    *
130 rem * c64          : ss=1024:se=2047:nc=198 *
135 rem ***************************************
140 rem * a good line to place the variables from above*
145 e%(0)=100 : e%(1)=82 : e%(2)=70 : e%(3)=64
146 e%(4)= 67 : e%(5)=68 : e%(6)=69 : e%(7)=99
150 print chr$(147)chr$(142); : rem * (clear) (graphics)
155 print " please choose the waveform desired " :print
160 print " (a) sine "
165 print " (b) cosine "
170 print " (c) logarithm "
175 print " (d) tangent "
180 print
185 input " your choice " ;c$ : c=asc(c$)-64
    : if c$<" a " or c$>" d " then 185
190 on c gosub 260, 265, 270, 275
195 input " step (fraction of 100) " ;sp:sp=sp/100
200 input " multiplier " ;mt
205 input " vertical additive (max 200) " ;ad
210 input " start " ;be
215 input " end " ;en
220 print chr$(147); : poke nc,0 : a=0 : d=int(ad/8)
    : e=ad-(d*8) : b=25-d
225 for xt=0 to sw-1 : poke ss+b*sw+xt, e%(e) : next
230 for xt=ss to se step sw : poke xt,79 : next
235 for z=be to en step sp : b=fn w(z)
    : if b>200 then b=201
240 d = int(b/8) : e=b-(d*8) : b=25-d
    : if sgn(b)=-1 then b=0
245 poke ss+b*sw+a, e%(e) : a=a+1
    : if a=sw then a=0
250 if peek(nc) then poke nc,0 : goto 150
    : rem *press any key to re-start *
255 next : print chr$(19); : end
260 def fn w(z) = abs(int(ad+mt*sin(z))) : return
265 def fn w(z) = abs(int(ad+mt*cos(z))) : return
270 def fn w(z) = abs(int(ad+mt*log(z))) : return
275 def fn w(z) = abs(int(ad+mt*tan(z))) : return
```

Mathematic Functions Expressed In Commspeak			Parameters
Secant	:	Sec(z) = 1/cos(z)	z<>pi/2
Cosecant	:	Csc(z) = 1/sin(z)	z<>0
Cotangent	:	Cot(z) = cos(z)/sin(z)	z<>0
Inverse Sine	:	Arcsin(z) = atn(z/sqr(–z*z + 1))	abs(z)<1
Inverse Cosine	:	Arccos(z) = –atn(z/sqr(z*sqr(–z*z + 1)) + pi/2	abs(z)<1
Inverse Secant	:	Arcsec(z) = atn(sqr(z*z–1)) + (sgn(z)–1)*pi/2	abs(z)>1
Inverse Cosecant	:	Arccsc(z) = atn(1/sqr(z*z–1)) + (sgn(z)–1)*pi/2	abs(z)>1
Inverse Cotangent	:	Arccot(z) = –atn(z) + pi/2	any value
Hyperbolic Sine	:	Sinh(z) = (exp(z)–exp(–z))/2	any value
Hyperbolic Cosine	:	Cosh(z) = (exp(z) + exp(–z))/2	any value
Hyperbolic Tangent	:	Tanh(z) = –exp(–z)/(exp(z) + exp(–z))*2 + 1	any value
Hyperbolic Secant	:	Sech(z) = 2/(exp(z) + exp(–z))	any value
Hyperbolic Cosecant	:	Cosh(z) = 2/(exp(z)–exp(–z))	z<>0
Hyperbolic Cotangent	:	Coth(z) = exp(–z)/(exp(z) + exp(–z))*2 + 1	z<>0
Inverse Hyperbolic Sine	:	Arcsinh(z) = log(z + sqr(z*z + 1))	any value
Inverse Hyperbolic Cosine	:	Arccosh(z) = log(z + sqr(z*z–1))	z>=1
Inverse Hyperbolic Tangent	:	Arctanh(z) = log((1 + z)/(1–z))/2	abs(z)<1
Inverse Hyperbolic Secant	:	Arcsech(z) = log((sqr(–z*z + 1) + 1)/z)	0<z<=1
Inverse Hyperbolic Cosecant	:	Arccosh(z) = log((sqr(z)*sqr(z*z + 1) + 1)/z	z<>0
Inverse Hyperbolic Cotangent	:	Arccoth(z) = log((z + 1)/(z–1))/2	abs(z)>1

Note: The "pi symbol" cannot be represented by all of the Commodore machines. If this is the case, use the value of pi = 3.1415926

Quarter Square Graphics

Another technique for obtaining pseudo high resolution graphics is through use of the quarter square graphics characters. In a single character space, there can be 16 different combinations of "quarters" – from completely blank (a space) to completely filled in (a reverse space) and every combination inbetween. This effectively doubles the resolution of your screen in both the vertical and horizontal directions.

This second program has been written by Paul Higginbottom. It was originally written for the 80 column machines but is easily adapted for 40 column or VIC 20. The setup subroutine at 9000 is where the changes will be made.

C64
Change : 9010 print " S "
: 9050 ln = 40 : bs = 1024 + 24*ln : . . .
Add : 9055 cs = 55296 + 24*ln
: 8025 poke cs + p, 1

Line 8025 is necessary for Kernal 2 C64s. POKing to the screen must be followed by a simultaneous POKE to colour memory or the characters will not show up.

VIC20
Change : 9010 print " S "
: 9050 ln = 22 : bs = 7680 + 22*ln : . . .
Add : 9055 cs = 38400 + 22*ln
: 8025 poke cs + p, 2

VIC20 with expansion
Change : 9010 print " S "
: 9050 ln = 22 : bs = 4096 + 22*ln : . . .
Add : 9055 cs = 37888 + 22*ln
: 8025 poke cs + p, 2

Since VIC 20 screens have only 23 lines, it will also be necessary to adjust the number 50 at the end of lines 410 and 430 to 46 (number of lines times 2). You will also nedd to change the first two numbers in the calculation for Y2 at lines 230 and 330. Start with:

Y2 = 23 + 22 * . . .

Warning: The plotting routine does not check to see if the POKE value is outside of screen memory. The potential for POKing into BASIC text space exists! Make sure you SAVE your program before trying new functions.

All the plotting efforts are performed by the two subroutines at 3000 and 8000. Subroutine 3000 plots a line from X1,Y1 to X2,Y2 by using subroutine 8000 to plot the points between the two coordinate pairs. When sub 8000 goes to plot a point, it must first determine if the target character space already contains a graphic. If it does, the new point must not interfere with the existing point in that space.

Lines 100 through 500 are some sample functions. The last one merely plots random lines end-–to–end. Of course any of these could be substituted by any of the mathematical functions shown in the table.

If you want to speed things up a bit, try substituting X for all occurences of X1 and Y for all occurences of Y1 in lines 100–500. Then change GOSUB 3000 to GOSUB 8000. This will simply plot the points instead of lines between the two coordinate pairs. There is a tradeoff which you will notice.

The variable N at lines 210 and 310 is the number of half cycles that will be displayed, and is automatically adjusted for screen size. To show more or less cycles, increase or decrease the first number (ie. 6*80/LN will show 6 half cycles or 3 complete cycles)

The decaying cosine wave at 300 will start at the top left of the screen (Y1 = 50). The decay is controlled by variable DC. For higher decay rates, use lower values in DC.

The last function at line 400 plots 5 random lines. With very little effort the same technique could be used to draw axes for the functions. The axes won't be disturbed by the waveforms thanks to the ingenuity of subroutine 8000. Simply set X1 = 0 : Y1 = 25 : X2 = 159 : Y2 = 25 and GOSUB 3000 for the horizontal axis. Then set X1 = 0 : Y1 = 50 : X2 = 0 : Y1 = 0 for the vertical axis.

Please write to us if simple variations can be made that would further increase the usefulness of these programs. RTE

```
50 gosub 9000
100 rem ********** functions **********
110 rem **** exponential curve ****
120 x1 = 0 : y1 = 1
130 for x2 = 0 to 159
140 y2 = exp (x2/42*80/ln)
150 gosub 3000 : y1 = y2 : x1 = x2 : next
200 rem ****      sine wave     ****
210 n = 6*80/ln : c = 3.1415926/180 : x1 = 0 : y1 = 25
220 for x2 = 0 to 159
230 y2 = 25 + 24 * sin(x2 * n * c )
240 gosub 3000 : y1 = y2 : x1 = x2 : next
300 rem **** decaying cosine wave ****
310 n = 8*80/ln : c = 3.1415926/180 : x1 = 0 : y1 = 50 : dc = 100
320 for x2 = 0 to 159
330 y2 = 25 + 24 * cos(x2 * n * c) * exp(-x2/dc)
340 gosub 3000 : y1 = y2 : x1 = x2 : next
400 rem **** draw 5 random lines ****
410 x1 = int(rnd(ti)*ln*2) : y1 = int(rnd(ti)*50)
420 for j = 1 to 5
430 x2 = int(rnd(ti)*ln*2) : y2 = int(rnd(ti)*50)
440 gosub 3000 : y1 = y2 : x1 = x2 : next
500 end
3000 rem ********** plot a line **********
3010 dx = x2-x1 : dy = y2-y1 : x = x1 : y = y1
3020 l = sqr(dx*dx + dy*dy) : if l = 0 then 3040
3030 xi = dx/l : yi = dy/l
3040 gosub 8000 : if (abs(x2-x)<=abs(xi)) and
     (abs(y2-y)<=abs(yi)) then return
3050 x = x + xi : y = y + yi : goto 3040
8000 rem **********  plot x, y  **********
8010 tx = int(x + ir):ty = int(y + ir) :sq = am(tx and am, ty and am)
8020 p = tx/dv-int(ty/dv)*ln : pokebs + p,c(i(peek(p + bs))orsq)
8030 return
9000 rem **********    setup    **********
9010 print " ssS " chr$(142);
9020 dim c(15), i(255), am(1,1)
9030 for i = 0 to 15 : read c(i) : i(c(i)) = i : next
9040 for i = 0to1 : for j = 0to1 : am(j,i) = (j + 1)*4↑i : nextj,i
9050 ln = 80 : bs = 32768 + 24*ln : dv = 2 : am = 1 : ir = .5
9060 data  32, 123, 108,  98, 126,  97, 127, 252
9070 data 124, 255, 225, 254, 226, 236, 251, 160
9080 return
```

Projectile Motion

Once you understand the techniques of putting objects on your screen, you'll want to get them moving. After all, what good is a sprite if it doesn't do anything. In this article we'll discuss some simple motion techniques using the laws of physics and mechanics.

Consider the screen of your computer as a 2–dimensional plane. To make an object move in 2 dimensions, you simply need supply a series of X and Y coordinates. Coordinate X usually represents horizontal position and Y is usually vertical position. Constantly changing the combination of these two positions will result in the illusion of motion. Calculating X and Y is a task determined by what pattern of motion you desire.

Calculating the path of a projectile can be done in one of two ways: the hard way and the easy way. The hard way would be probably end up as a collage of imaginative calculations that somehow produce a fairly accurate simulation. The easy way is the logical way. In any book of physics or mechanics you'll find just about every formula for plotting the path of an object that is directly affected by a forward velocity, an upward velocity, and gravity – inotherwords, a projectile.

Forward Velocity

Every moving object on Earth has a forward velocity. Even if it only goes straight up, then straight down, it has a forward velocity. Of course this would be a forward velocity of zero.

Velocity is represented as some unit of distance, per some unit of time (eg. 10 feet/second). On the computer, the units of distance will be a column on the screen.

The units of time could be obtained from the internal clock, but this imposes certain unnecessary complications. First, the lowest unit is seconds which is an awfully long time unless the velocity is very low too. Second, when the seconds reach 59, it is up to the programmer to add the minutes times 60.

The solution is to simulate time with a simple FOR/NEXT loop. This way you can express time in any unit such as tenths of seconds or even 3rds of seconds if you wish. Also, this avoids the potential for losing time since the clock will not increment until you have used the current time for your calculation, and subsequently used the results of that calculation for the plot. So far our formula will look like:

```
100 fv = 10 : rem forward velocity
120 for t = 0 to 159 step .2
130 x = fv * t
160 gosub 8000 : rem plot a point
180 next
```

The subroutine at 8000 is from Paul Higginbottom's plotting routine in the previous article. Note: please see the program at the end of the article – the programs presented in this text are meant to show the progression to the final program.

As you can see, time will be incremented from 0 to 159 in steps of 0.2, simulating a fifth of a second clock. The X coordinate is calculated and delivered to a subroutine that plots the coordinate on the screen. This subroutine could be replaced by one that changes the coordinate of a sprite.

But no upward velocity has been given. In this case the projectile will simply move horizontally until the clock stops.

Upward Velocity and Gravity

This is the next element of the path of a projectile. It too is represented in distance per time unit, but unlike forward velocity, it is affected by the phenomena of Gravity.

Gravity is a unit of acceleration. When you drop an object, it starts with a velocity of zero and accelerates. Gravity is usually given as –32.2 feet per second squared. Different locations on Earth have gravitational constants slightly different than this depending on height above/below sea level, etc., but we'll use the natural constant for now. Further, if you go up high enough to drop your object, it will accelerate to a maximum velocity of about 119 mph, but we won't be doing that either.

The formula for our Y coordinate becomes the upward velocity multiplied by time, minus the effect of gravitational pull:

$$Y = UV*T - {}^1/_2G * T^2$$

The program becomes:

```
100 fv = 10 : rem forward velocity
110 uv = 45 : rem upward velocity
120 for t = 0 to 159 step .2
130 x = fv * t
140 y = 5 + uv * t – .5 * 32.2 * t↑2
160 gosub 8000 : rem plot a point
180 next
```

The value 5 at the beginning of line 140 is an initial height so that the projectile does not start from 0 elevation. This brings us to the next consideration.

Impact and Decaying Velocity

As you well know, what goes up must come down. When our object hits the Earth, it will bounce, unless our object is made of

wet cement. Upon impact the object loses some of its initial upward velocity. Technically this is referred to as decay or a damping factor. However, our program poses another problem in its present form.

After the object impacts, it will want to continue back up. Logically, time has become zero again and the formula repeats itself at the new decayed upward velocity. But our program shows time always incrementing. Therefore we must have a method of resetting the clock when the new cycle begins. The FOR/NEXT loop is the target for our next modification. Instead of representing time, it will now simply act as a counter, and time will be calculated elsewhere.

We need also know when the object impacts. Inotherwords, where is ground. Since we started from an initial height of 5, we'll say that ground is at 5. So when our calculation for Y yields a result less than 5, we know the object has bounced. This is also the point at which the decay takes effect. In this example, only upward velocity will decay – the forward velocity of a bouncing object is shown to be fairly constant, although you could impose damping on it too if you wish. The program becomes:

```
100 fv = 2  : rem forward velocity
110 uv = 45 : rem upward velocity
120 for j = 0 to 159
130 x = fv * j
140 y = 5 + uv * t - .5 * 32.2 * t↑2
150 if y<5 then y = 5 : t = 0 : uv = uv*.9
160 gosub 8000 : rem plot a point
170 t = t + .2
180 next
```

Notice that FV has been changed to 2 in line 100. This simply allows more cycles on the screen to show the effect of impact. Line 150 says if Y is less than 5 then Y equals 5. This little bit of cheating makes the ball bounce at the same vertical spot on the screen each time. Time is reset to zero and upward velocity is reduced by 10 percent.

Summary

With the program now in its final form, several possibilities exist. You can vary gravity slightly to show the effects of impact at different spots on Earth, or vary it a lot to simulate gravity on other planets. Your starting point does not necessarily have to be the same as the point of impact when the object comes down – you might project your object from some much higher elevation (eg. a cliff). Also, the object might not be the bouncing type – bombs don't usually bounce. If your object is the type to bounce, try different decay values for objects made from different materials. Depending on how hard they bounce might affect the forward velocity too – something to think about.

Remember one thing most – computer simulations are all too often a task of logical thinking. The actions and reactions of an actual physical object can usually control the computer to demonstrate these actions, and thus obtain the desired result. Think first, program later.

Portability Notes

The program has been set up for 80 column machines. Line 9010 prints two HOMEs to clear any windows, a clear screen, followed by a set graphics mode – CHR$(142). For 40 column PET/CBMs simply change LN = 80 to LN = 40 in line 9050. If you have entered Paul Higginbottom's quarter square graphics routine from the previous article, you only need to add lines 500 to 590 below. Notice how only the subroutine at line 8000 is called – subroutine 3000 has been omitted here. Commodore 64 and VIC 20 users will need to make the same changes noted in the previous article.

As it stands, the program will leave a trail along the path of the projectile. To remove the trail, add:

```
8005 poke bs + p, 32
```

Line 8005 will poke a space into the previous POKE position thus erasing whatever was there. This is all that is required for all machine models.

Question: How would you make the ball bounce off the right hand edge of the screen (ie. a wall). Hint: if x>79 then x = 160–x. Use FV = 7:UV = 55.

```
50 gosub 9000
500 rem ****    bouncing ball    ****
510 fv = 2 : uv = 55 : y1 = 1 : g = -32.2 : dc = .9
515 rem try fv = 15 : uv = 45. also fv = 0.2
520 for j = 0 to (ln*2-1)/fv
530 x = fv*j
540 y = y1 + uv * t + .5*g * (t↑2)
550 if y<y1 then y = y1 : t = 0 : uv = uv*dc
560 gosub 8000 : rem plot a point
570 t = t + .2
580 next
590 end
8000 rem **********  plot x, y  **********
8010 tx = int(x + ir):ty = int(y + ir) :sq = am(tx and am, ty and am)
8020 p = tx/dv-int(ty/dv)*ln : poke bs + p,
       c(i(peek(p + bs))orsq)
8030 return
9000 rem **********    setup    **********
9010 print " ssS " chr$(142);
9020 dim c(15), i(255), am(1,1)
9030 for i = 0 to 15 : read c(i) : i(c(i)) = i : next
9040 for i = 0to1 : for j = 0to1 : am(j,i) = (j + 1)*4↑i : nextj,i
9050 ln = 80 : bs = 32768 + 24*ln : dv = 2 : am = 1 : ir = .5
9060 data 32, 123, 108, 98. 126, 97, 127, 252
9070 data 124, 255, 225, 254, 226, 236, 251, 160
9080 return
```

Voice For Commodore Computers

Vikash Verma
Calgary, Alberta

The idea of voice communication with the computer is fascinating, to say the least. There are many ICs in the market today which can synthesize human speech such as the Votrax SC–01 and those in Texas Instrument's Speak & Spell.

There are two aspects of voice communication – listening and speaking. The above ICs accomplish only one of these activities – speaking. Listening, which is also known as speech recognition, is far more complex. Nevertheless, microcomputer based speech recognizers such as Voiceteks' Cognivox and Voice Input Module sold by MCE, Inc. are also available in the market, albeit at higher cost.

The present article deals only with interfacing one of the speech synthesis devices marketed under the name "Sweet Talker" by Micromint, Inc. (561 Willow Ave, Cedarhurst, NY, 11516, 516–374–6793). This device is based on the Votrax SC–01 chip. In this article Sweet Talker will be interfaced with the User Port of the CBM machines to add a new dimension to games and educational programs.

The voice response from the computer in human speech can improve certain games enormously and can open new avenues for adding humour to games or zip to educational software. Pre–school children who cannot yet read, can interact with the computer through voice much better than through written instructions on display CRT. Computers can speak out the instructions such as "Press Y to start the game" to a child who cannot yet read such sentences from the screen.

Hardware

Sweet Talker requires dual power supply voltages of +5V and +12V. The +5V on the cassette port (pin 2 or B) can be used for one. Pin 4 or 5 of J10 in the PET/CBM can be used for the +12V supply but VIC 20/C64 users will need an outside alternative.

The fundamental phonetic alphabet, called phonemes, are transferred to the speech synthesizer on six parallel lines P0–P5, allowing 64 possible phonemes. Micromint supplies a list of the 64 phonemes with Sweet Talker. There are two pitch control lines, I1 and I2, and three more connections, STB, A/R, and Enable, for handshaking purposes. Each phoneme is uttered by Sweet Talker for a pre–determined duration varying from 40 ms. to 250 ms. Figure 1 shows the interfacing of a CBM with Sweet Talker. Leaving pins I1 and I2 unconnected causes the built–in random pitch variation to be in effect.

Software

The program "Talking Clock" shown in listing 1 was developed for PET/CBMs (see VIC 20 and C64 notes below). It demonstrates how the hand–shaking is performed to communicate with the speech synthesizer.

The program sounds off the time of day once every minute, or by pressing any key at any time. Lines 200 and 210 allow you to synchronize the internal clock with the present time of day. Line 150 prepares the pin CA1 of the computers' User Port to detect a negative transition (voltage going from high to low) sent by Sweet Talker.

The heart of the program is the subroutine from 440 to 500 which sends the 6 bit code of the phoneme to be pronounced and waits until the phoneme enunciation is complete. Line 440 place the phoneme code on the data output lines. Lines 450 through 470 send a pulse to Sweet Talker via CB2 to indicate that a valid phoneme code is available on the data lines. Sweet Talker then begins to produce the sound corresponding to the particular phoneme code. After a pre–determined duration, Sweet Talker drops line STB connected to pin CA1 from a high to low voltage to indicate that it is ready for the next phoneme. The computer waits for this negative transition in line 490. The DATA statements 510 through 760 contain the ASCII equivalent of phoneme codes of the various english words required for this demonstration.

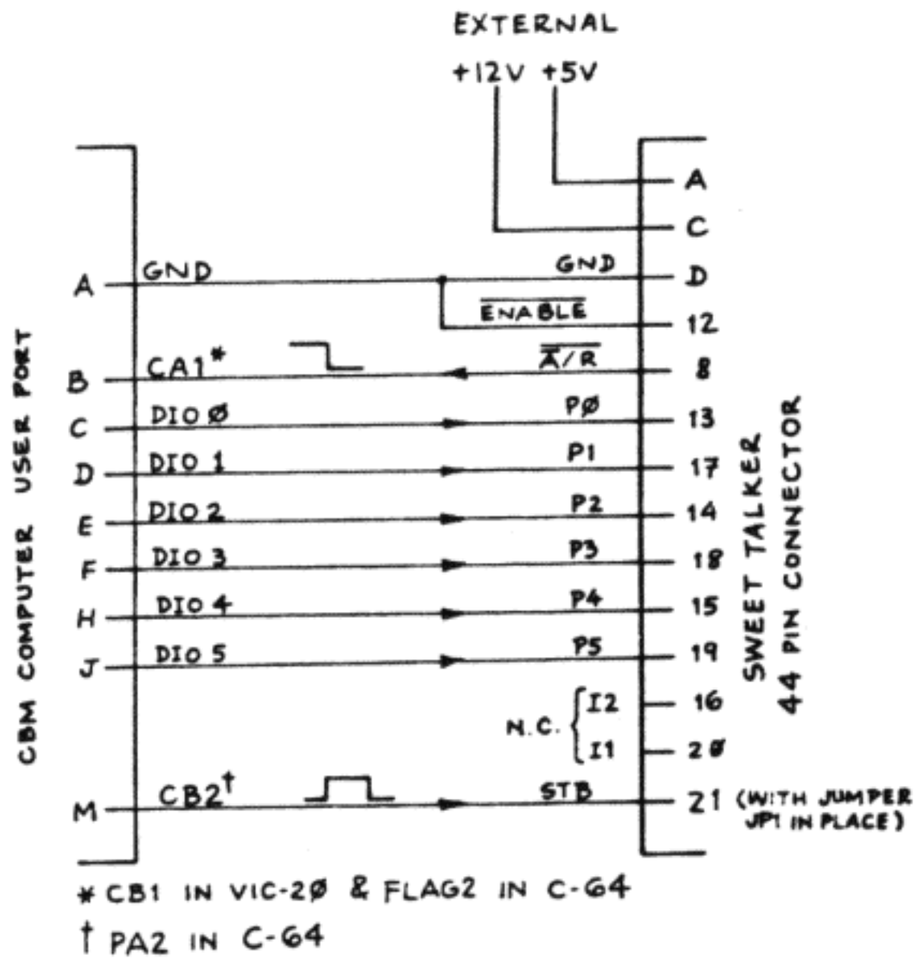


FIGURE 1. CBM/SWEET TALKER INTERFACE

VIC 20 Notes

In line 140, RO and PC should be 37136 and 37148 respectively. In line 150, 254 should be replaced by 239. In line 90, SP should be assigned a value of 7, and line 490 should be replaced by:

```
490 if (peek(fr) and 16) <> 16 then 490
```

The changes in lines 150 and 490 are due to the fact that on the VIC 20 User Port, CB1 has been made available in place of CA1.

Commodore 64 Notes

Replace lines 140 and 150 by the following:

```
140 ra = 56576 : rb = ra + 1 : da = ra + 2 : db = rb + 2
    : ic = ra + 13
143 poke db, 255 : rem set port b for all outputs
146 x = peek(ic) : rem dummy read to set icr
150 poke da, peek(da) or 4 : rem set pa2 for output
```

and replace lines 440 through 490 by the following:

```
440 poke rb, x : rem send ascii of phoneme to sweet talker
450 rem send one shot pulse on pa2
460 da = peek(ra) : poke ra, da or 4
470 poke ra, da
480 rem wait for neg transition on flag2 (pin 8)
490 wait ic, 16
```

References

Ciarcia, S. 'Build an Unlimited-Vocabulary Speech Synthesizer', BYTE Sept 1981, pp 38-50

MOS Technology Inc. MCS6522 Versatile Interface Adapter Data Sheet, March 1977, 24 pages.

Talking Clock

```
90 c1$ = chr$(17) : c2$ = chr$(147) : sp = 16
100 printc2$c1$ "    t a l k i n g  c l o c k"
110 printc1$ "    (c) vikash verma,calgary"
115 printc1$c1$ "this clock will sound off";
120 print " the time of day"
125 printc1$ "once every minute" : printc1$c1$ "    or"
130 printc1$ "whenever any key is pressed"
135 cr$ = chr$(19) : fori = 1to10 : cr$ = cr$ + chr$(17) : next
136 fori = 1tosp : cr$ = cr$ + " " : nexti
140 ro = 59457 : pc = 59468 : dd = ro + 2 : fr = pc + 1
    : pokedd,255 : rem set ora for output
150 pokepc,peek(pc)and254 : rem set pcr to detect -ve
    transition on ca1
160 dimb$(20),c$(5) : d$ = ">>" : am$ = "f!)>ball"
    : pm$ = "%<)>ball"
170 reada$,e$
180 fori = 0to20 : reada$,b$(i) : nexti
190 fori = 2to5 : reada$,c$(i) : nexti
200 printc1$c1$c1$c1$c1$c1$ "   to set clock enter time
    in the format : hhmmss"
210 inputti$ : printc2$;
220 t$ = ti$ : printcr$;left$(t$,2) ":" mid$(t$,3,2) ":";
    mid$(t$,5) : t$ = left$(t$,4)
230 mr$ = am$
240 hr = val(left$(t$,2)) : mn = val(right$(t$,2))
250 geta$ : ifa$<> "" ort$<>ts$thents$ = t$ : goto270
260 goto220
270 a$ = e$ : gosub410 : ifhr = 0thenhr = 12 : goto300
280 ifhr>12thenhr = hr-12 : mr$ = pm$
290 ifhr = 12thenmr$ = pm$
300 a$ = b$(hr) : gosub410 : ifmn = 0goto370
310 ifmn<10thena$ = b$(0) : gosub410
320 ifmn<=20then340
330  m1 = int(mn/10)  :  mn = mn-m1*10  :  a$ = c$(m1)  :  go-
sub410
340 a$ = b$(mn) : ifmn>0thengosub410
350 a$ = mr$ : gosub410
360 goto220
370 ifhr<12then350
380 ifmr$ = am$thenmr$ = "li↑mh@)*"
390 ifmr$ = pm$thenmr$ = "m(m"
400 goto350
410 a$ = a$ + d$ : for a = 1 to len(a$)
420 x = asc(mid$(a$,a,1)) : gosub 440
430 next a : return
440 poke ro, x : rem send ascii of phoneme to sweet talker
450 poke pc, peek(pc) or 224
460 rem send one shot pulse on cb2
470 poke pc, (peek(pc) and 31) or 192
480 rem wait for -ve transition on ca1
490 if (peek(fr) and 2) <> 2 then 490
500 return
510 data the time is, 83>*h@)l>kr>>
520 data o, 444
530 data one, -21m
540 data two, *(
550 data three, 9+<)
560 data four, ]55+
570 data five, ]h)o
580 data six, ←iyc←
590 data seven, ←@o@m
600 data eight, e)*
610 data nine, mh)m
620 data ten, *am
630 data eleven, jxeo@m
640 data twelve, *-axoc
650 data thirteen, "9:+*,m"
660 data fourteen, "]5+*,m"
670 data fifteen, "]i]*,m"
680 data sixteen, "←iyc←*,m"
690 data seventeen, "←@o@m*,m"
700 data eighteen, "e)d,m"
710 data nineteen, "mh)m*,m"
720 data twenty, "*-am*,"
730 data thirty, "9:+*)"
740 data forty, ]5+*)
750 data fifty, ]i]*)
```

Hardware Corner

Domenic Defrancisco
Chris Zamara
Downsview, Ont.

Welcome to Hardware Corner. This is the first in a series of articles written to help you learn about the input/output capabilities of your PET, VIC, or C64. In this article, we will look at the user port, and how to use the chip that controls this port from BASIC. In future articles, we will build up on this information, and connect things up to the computer, such as LEDs, seven segment displays, stepping motors, and other assorted lights, bells, and whistles.

Introducing The USER PORT

The User Port, physically, is a small rectangular slot on the rear of your computer which accepts a standard edge card connector. Coming from the connector are a number of lines. These lines carry electrical signals which you can control via software. The user port lines can be accessed from BASIC via PEEK and POKE just like any other memory location. This PEEKing and POKEing controls an I/O chip in the computer which is connected to the user port. The chip is a 6522 in the PET and VIC, and a 6526 in the C64.

This accessing scheme is known as MEMORY MAPPED I/O, and is used in controlling everything external to the CPU, such as the video, sound, keyboard, joysticks, etc. By POKEing values into certain memory locations, the user port can be set up to operate in different modes, and data can be sent out on the port. By PEEKing locations, the current state of the port can be read. By using the port in this way, the computer can control or communicate with a variety of external devices, which may be as simple as a row of LEDs, or as complex as a disk drive, or even another computer.

On the user port, there are a number of different types of lines available. The most flexible of these lines are those belonging to the PARALLEL PORT. The parallel port consists of 8 lines, labelled 0–7, each one individually usable as either an input or an output. When a line is used as an output, the computer can make that line either low (zero volts), or high (+5 volts). When a line is used as an input, the computer can sense whether the line is being held high or low by an external device. This is the essence of communication between the computer and any other peripheral. Besides the lines of the parallel port, there are lines which control the transfer of information (handshaking lines), and other lines performing complex functions such as serial data transfer, and timing

The PARALLEL PORT

First of all, the parallel port. How do we use it? In the preceding paragraph it was mentioned that each of the 8 lines may be independently used as an input or an output. To tell the I/O chip which lines are inputs and which are outputs, a value is POKEd to the DATA DIRECTION REGISTER. The data direction register for the parallel port is location 56579 on the C64, 59459 on the PET, and 37138 on the VIC (you can find these register locations, along with other important data about the user port, in Transactor's reference issue). Each line on the user port corresponds to a bit in the data direction byte. If a bit is set to zero, the corresponding line on the port is an INPUT. If the bit is set to a one, the corresponding line is set as an OUTPUT.

For those of you not familiar with binary, see figure 1. For each line that is to be an output, look up the corresponding number in the right hand column of the chart. Add up all of these numbers to get the value to POKE into the data direction register. For example, if all lines are to be outputs, add:

$$1+2+4+8+16+32+64+128=255$$

Bit Values

Bit	Dec	Hex
0	1	$0001
1	2	0002
2	4	0004
3	8	0008
4	16	0010
5	32	0020
6	64	0040
7	128	0080

Figure 1

In the C64, we would POKE 56579,255 to set all the lines as outputs, and we could then change the level of all lines on the parallel port. If we wanted lines 0 through 3 to be outputs, and lines 4–7 to be inputs, we would add: 1+2+4+8=15, and we would use POKE 56579,15. In this configuration, we could change the levels of lines 0–3, and read the state of lines 4–7.

Now that we have set up the data direction register, we can control the port itself through the DATA REGISTER. This register is located at 59471 on the PET, 37136 on the VIC, and 56577 on the C64. The bits in this register also correspond to the lines on the parallel port, but a '1' corresponds to a high voltage level on the port, and a '0' corresponds to a low level. This means that if we had previously set line 3 as an output using the data direction register, and we POKEd the value 8 into the data register (the corresponding value from figure 1), line 3 on the port would have 5 volts present on it. Likewise, an input line could be read via a PEEK to detect whether it is being held high or low by an external device. To find out whether an input line is high, PEEK the data register, then AND the contents with the corresponding value for the desired line from the table. To make the above procedures more clear, examine the following short BASIC program, written to work on the C64 (to use on another machine, just change the data direction register and data register locations in lines 20 and 30).

```
10 rem ** parallel port example for c64 **
20 ddr = 56579 : rem data direction register
30 dr  = 56577 : rem data register
40 poke ddr,31  : rem set lines 0-4 as outputs,
lines 5-7 as inputs
50 poke dr,1    : rem make line 0 high, 1-4 low
60 p = peek(dr) : rem examine port
70 if p and 32 then print "line 5 high"
80 if p and 64 then print "line 6 high"
90 if p and 128 then print "line 7 high"
```

The above program first sets the data direction register, POKEs to the data register to set line 0 high and lines 1–4 low, and then reads the port and reports which of the input lines 5–7 are high. This is a simple example of how to use the parallel port, and in future articles we will be connecting things up to the port, so the above program might turn on an LED, and then report the state of three toggle switches.

And The Other Lines

The parallel lines described above will be used in almost all upcoming projects, but the other, more complex control lines which are available should also be mentioned. Below is a brief description of the lines available on the PET, VIC, and C64 computers. A more detailed description of the lines, and the method of using them will be given as they are used in future installments.

Control Lines Available On The PET's User Port

CA1 : An input which sets a flag on an input pulse, and can be programmed to generate an interrupt when the flag is set, or latch the contents of the parallel port.

CB2 : As an input, can set a flag like CA1. As an output, can be used as a programmable shift register. When used as a shift register, CB2 can be used for serial data transmission, or for generating sound through a speaker.

Control Lines On The VIC 20

CB1 : Similar to CA1 in the PET
CB2 : same as CB2 in the PET
CASSETTE SENSE : Detects whether the cassette PLAY button is pressed. If no cassette is connected, can be used as an input or output.
JOY 0 – JOY 2 : Three of the four input lines from the joystick port (can be used as outputs if no joystick is plugged in).

Control Lines On The Commodore 64

FLAG : similar to CA1 in the PET
PC : Indicates when data on the parallel port is valid. Often used in conjunction with FLAG for parallel data communication.
CNT1, CNT2 : Inputs used in conjunction with internal timers to measure input frequencies and pulse widths. Also aids in serial communications when using the SP line.
SP : Used for serial transmission.
PA2 : An extra I/O line from parallel port A.

Next Issue. . .

Now that we have established some background theory about what the user port is and how to control it from BASIC, we are ready to start really using it! In the next issue, we will look at how to physically connect circuits to the user port, and use a row of 8 LEDs as an example.

How Cartridges Work

Daniel Bingamon
Batavia, Ohio

Have you ever wondered how a Commodore 64 cartridge starts up a game automatically without you giving any commands? The way it is done is fairly simple if you know a little machine language. Here we are going to show you how to use this cartridge control to make the computer start up with different colors.

When you start up your computer or reset it, it must have a place to go on start. This starting point is determined by the RESET vector. Upon reset the 6510 loads the program counter (a place in the microprocessor the computer uses to know were it is at) with the address that is in the reset vector. The 6510 "knows" that the reset vector is stored at $FFFC and $FFFD in ROM. Its contents is address $FCE2.

The microprocessor transfers execution to $FCE2, the RESET Routine. One of the first tasks performed is a subroutine at $FD02 that tests for the presence of a cartridge. This test compares the characters 'CBM80' to the contents of locations $8004 through $8008. If 'CBM80' is present and the Most Significant Bit is set to one in the letters 'CBM', then the KERNAL ROM does an INDIRECT jump through o location $8000 (32768 in decimal). Locations $8000 and $8001 are the first two bytes of the cartridge and are used to store the address of the start of the cartridge program to give the cartridge control.

The next two bytes, $8002 and $8003, are the NMI Reset address. This is the address the 64 will jump to if RUN/STOP–RESTORE is pressed, but again, only if 'CBM80' is present at $8004.

Cartridge Simulation

Because there is RAM at location $8000 (unlike the VIC–20 which is normally blank in its cartridge space), we can fool the C64 into thinking it has a cartridge plugged in. When there is a real cartridge in the computer, wiring is provided to turn off the RAM and connect the cartridge instead. This program will not work if you have a cartridge in the computer.

If you look at this program with a machine language monitor, notice the start–up code 'CBM80' at $8004. It must show exactly like it is listed here in order to gain control of the computer:

```
.: 8000 09 80 2f 80 c3 c2 cd 38
.: 8008 30 Start Of Program. . .
```

The space marked 'Start Of Program' is where the cartridge program starts. In general, a cartridge program could start anywhere in the cartridge, but the program in our simulated cartridge will start here. Now when the computer is reset, it will go to location $8009 and execute that program. There better be something there or you will probably get a non–recoverable crash, and turning off the computer will fix things – a reset button may not get you out of this one.

The Start Of Program begins with some initialization routines just like the ROM reset routine. First, the screen initialization at $FF81 puts the screen in the right place. After that is done a call is made to $FF84 which initializes I/O devices. Some cartridges never use this, but I find things run much smoother when you do use it. The next thing this program does is execute the routine called 'CLRCHN' to clear I/O channels and its address is $FFCC. After all this is done I store the new colors to the graphics chip and do an indirect jump through $A000 which starts up BASIC (ie. power up message, bytes free, etc.). However, this jump would be replaced by the cartridge program which would continue on from there. With this technique you can test your cartridge program in RAM, which later you can have 'burned in' on cartridge.

Also included in this program is a section that sets the NMI vector to change your screen color to the new default upon hitting the restore key and continue running any BASIC programs without stopping them, although the run/stop key by itself still works.

Listed with this article is the BASIC loader program that pokes this machine language into memory. You will be notified by the program if you have mistyped any DATA. To make this machine language to take effect type 'SYS64738' and upon restart you will notice something different. Please SAVE this program before typing the SYS statement.

Use SUPERMON or any other equivalent monitor to follow the flow of this program. Please type 'POKE 56,128:CLR' to protect the program from strings writing over it.

Program Listing:

```
5 printchr$(147) : print " cartridge simulator "
6 print : print " by:daniel bingamon
10 for i = 32768 to 32838
20 read a : b = a + b : poke i,a
40 next
50 if b<>7693 then print " data error " : end
100 data    9, 128,  47, 128, 195, 194, 205
110 data  56,  48,  32, 129, 255,  32, 132
120 data 255,  32, 138, 255,  32, 204, 255
130 data 169,  18,  32,  56, 128, 169,  46
140 data 141,  24,   3, 169, 128, 141,  25
150 data   3, 108,   0, 160,  72, 152,  72
160 data 138,  72,  32, 204, 255,  32,  56
170 data 128, 104, 170, 104, 168, 104,  64
180 data 169,  11, 141,  32, 208, 141,  33
190 data 208, 169,   0, 141, 134,   2,  96
200 print:print " sys64738 will begin simulation "
```

Generator Programs

Jim Butterfield
Toronto, Ont.

A "generator" program is a program that writes another program. This can open up numerous possibilities: tell a generator program what you want, and it will write the program for you. I'll describe a much simpler generator program: one that allows easy, error–proof entry.

The previous Transactor contained a program to copy files on the Commodore 64 with a 1541 disk. Much of the program was in machine language. Some users who might find the program handy don't know how to enter the hexadecimal code that comes with machine language. Even those who do have another problem: one mistake can blow the whole program. I decided to write a generator program to help with these problems.

Objectives

The "easy" way to get machine language into a computer is to encode it into DATA statements. That way, the user simply types in Basic statements without having to learn how to use the machine language monitor. So far, we haven't checked for errors.

We can try to guard against errors by introducing a checksum system. The simplest checksum would work this way: add all the numbers in the DATA statements together, and check to see that the total matches some sort of correct sum. That's OK for small programs; but bigger ones, which are likely to have several entry errors, would call for the user to proofread everything over and over again until the checksum matches. It would be better to have a scheme where a specific data line is signalled. In this case, of course, each line must be given its own checksum.

There's another problem with checksums: it's too easy for errors to "balance out". A simple transposition, for example, with the user typing 35,62 instead of 62,35 would produce the same total either way; the checksum wouldn't catch the error.

Checksums can introduce other pitfalls, too. Suppose that we have a separate checksum for each line of data. If a line is omitted, this may not be spotted – it can't have a checksum error if it's not there. One other thing: I wanted to avoid a scheme that would force an exact number of data elements per line. If you have, say, a 631–byte program it would be hard to fit it into an number of DATA statements each containing the same number of elements.

First Planning

DATA statements should start at line 1 and go consecutively from there. Each DATA line should be separately checksummed. If an error occurs, the DATA line with the error should be identified.

The checksum should operate along the following lines. It should include the DATA line number, so that we'll spot the problem if a line is incorrectly entered with the wrong number. It won't be a simple addition – if we multiply the total by three before adding a new value, each will have a different weight. To ensure the checksum doesn't become too large and unwieldy, we'll trim it with an AND of 63. And we'll set the checksum aside by making it a negative number.

Where do we start on all this? The code started out along the following lines:

```
L = PEEK(63) + PEEK(64) * 256
```

This gives us the DATA line number. New rule: if we make sure all DATA lines are lower than 256, we can shorten this to:

```
L = PEEK(63)
```

By the way, location 63 is valid for VIC and Commodore 64; the proper value would be 60 for PET/CBM. It might be easier to set variable M to 63 and use PEEK(M); that would make the program easy to modify for PET.

Now, the last line of DATA will be line 200, and it will contain a value saying how many lines of DATA, plus one, are in the main sequence. So we will do a boolean check on the line number, L:

```
H = (L = 200)
```

H will be "true" if we have reached DATA line 200; otherwise it will be false. We know that the following program will be looking to see if the lines are in the right sequence. Line

200 will be out of sequence, of course, because it has a special meaning. So, when we reach line 200 (with H "true" we'll arrange to pass the sequence test by changing the 200 to the expected next line, with:

```
IF H THEN L=X
```

A value which is "true" or "false" is called a boolean variable. We're going to create a couple more. Variable V will tell us if we have moved to a new line of DATA; it checks the line number, L, against the previous line number, R. If they are the same, we're on the same line as before. If not, we've moved to a new line:

```
V=(R<>L)
```

V will be "true" if we have just moved to a new line. If so, we want to check the line we have just left to see if there are any checksum errors. S is a boolean variable signalling an error; it is set only when end–of–line is true. Thus, we know that S depends on V, new–line, and that we must write something like S=(V AND. . ..).

When we hit the first line, we don't want to check the previous line for errors . . . there is no previous line. So we'll insist that S, the error flag, can be set only when the previous line number (R) is greater than zero. This takes us to S=(V AND R>0 . . .).

If all other conditions are satisfied – new line, and not the first – we actually check to see that the checksum is correct. The arithmetic is set up so that the checksum should always work out to be 63. So our checksum error boolean S is set with:

```
S=(V AND R>0 AND T<>63)
```

S is "true" if and only if there has been a checksum error on the previous line. We won't print the error statement yet.

If we have moved to a new line (V is "true"), we'll set the checksum value to beequal to the line number; as we go through the line's DATA statements, this will be mixed in with other values:

```
IF V THEN T=L
```

If this is a new line (V is true) and there is no error (S is false) we need to make one more check: is the line number in sequence? Here's how we do this: we add one to R (the previous line number), and check to see that the new value is equal to L, the new line number. If it does not, set S to true and flag an error.

```
IF V AND NOT S THEN R=R+1:S=(R<>L)
```

Now, let's do the checksum calculation. T is the checksum total; each time we read a chunk of DATA, we multiply the previous value of T by three and add the new DATA item; then we trim the result to the range 0–63 with an AND statement:

```
T=(T*3+X) AND 63
```

Now we're ready to report an error, if S is true:

```
IF S THEN PRINT "ERROR LINE";R
```

Note that we don't print line L, the new line we have just reached. Instead, we print R, the previous line number; or if a line is missing, the previous line number plus one. If we have an error, we'll also record this into variable E:

```
IF S THEN E=-1
```

A value of –1 is equivalent to logic "true", by the way; so E is a boolean and will become true and stay that way if S is ever true. We could alternately write E=S OR E and even drop the spaces to write E=SORE, but the above coding works equally well.

Now we make our "previous line number", R, equal to the current line number, L; and if we are not at line 200 (that's indicated by variable H, remember?) we may go back to do the next data READ:

```
R=L:IF NOT H GOTO. . . read statement
```

If we don't go back, we have read all our DATA statements, and boolean variable E will tell us if any of them had errors. If so, we stop so that the errors can be fixed:

```
IF E THEN STOP
```

From here on, we go back to the beginning of the DATA statements with RESTORE and read them all in again. This time, we know they are correct and can put them to work – or to the disk – right away. We skip the negative values: they are the checksum.

Disk or Memory?

The program given in the Transactor wrote a disk program called COPY FILE. That program, once written, could be loaded and run.

That's the handiest way for users who have a disk system. For users without a disk, there's a trickier method which allows a program to "collapse" upon itself in memory, generating a new program which may then be saved to disk or tape.

Demonstration Program

Here's a simple example of a "collapsing" program. It's written specifically for the Commodore 64 – don't try it on a VIC or PET/CBM.

It collapses into a three–line Basic program. Big deal; it doesn't do anything amazing for Basic. But it's a Basic program you'd have trouble typing in the regular way.

```
1 data 27,8,100,0,83,36,178,34,144,5,-25
2 data 28,159,156,30,31,158,129,149,150,151,-42
3 data 152,153,154,155,34,0,141,8,110,0,-29
4 data 153,32,200,40,83,36,44,187,40,49,-53
5 data 41,172,49,54,170,49,41,59,34,147,-26
6 data 17,17,32,32,32,32,84,72,73,83,-37
7 data 32,73,83,32,32,141,32,32,65,32,-44
8 data 68,69,77,79,78,83,84,82,65,84,-36
9 data 73,79,78,32,141,32,32,32,32,32,-37
10 data 79,70,32,65,32,76,73,78,69,32,-23
11 data 141,32,32,32,32,84,72,65,84,32,-18
12 data 89,79,85,32,32,67,79,85,76,68,-15
13 data 78,39,84,32,141,32,32,32,32,32,-44
14 data 32,32,84,89,80,69,32,73,78,0,-2
15 data 151,8,120,0,137,32,49,49,48,0,-18
16 data 0,0,-49
200 data 17
300 m=63
310 read x:l=peek(m):h=l=200:if h then l=x
320 v=r<>l:s=(t<>63 and r>0 and v)
330 if v then t=l:if not s then r=r+1:s=r<>l
340 t=(t*3+x)and63
350 if s then print "error line";r:e=-1
360 r=l:if not h goto 310
370 if e then stop
380 print"here we go":x=-1 : restore : b=2049
    : for a=1 to 9999
390 if x>=0 then poke b,x:b=b+1
400 read x:l=peek(m):if l<255 then next a
410 poke 45,16:poke 46,16:clr
```

Type it in. Try creating deliberate mistakes in the data lines; it would be most unusual for it not to be caught by the checking program. When everything is OK, the program will collapse into a three–line Basic program which can be saved to tape or disk in the usual way.

Don't worry too much about the strange screen you get when you say LIST. You can't enter this program in the usual way because the second line contains special characters that cannot be entered or listed.

Data line 16 is sufficiently short that you can work through the checksum by hand. Start with the line number, 16. Multiply by three and add zero giving 48. Multiply by three again and add zero: this gives 144 which when ANDed with 63 gives 16 (divide by 64 and take the remainder). One last time: 16 times 3 is 48 plus –49 gives –1, which works out as 63 when ANDed with 16. And that's the correct checksum.

A Generator Generator

I worked out the data lines for the COPY program semi–manually, presenting the data on the screen and entering it by moving the cursor over and pressing RETURN. That's tedious for medium to large programs.

Elizabeth Deal wrote a data line generator for this program before I did. It needs a disk; information is drawn from a program (called "INPROG" here) and a new program (called "OUTPROG" here) is produced which contains the DATA statements. The following listing is adapted from the original Deal program.

```
2030 z$=chr$(0):d$=chr$(131)+"  ":d=10:ll=1
2050 open 1,8,3,"0:inprog,p,r"
2060 open 2,8,4,"0:outprog,p,w"
2070 print#2,chr$(1);chr$(4);
2080 for l=1 to 1e4:gosub 2170
2090  for k=1 to d
2100   get#1,i$:v=asc(i$+z$):ef=st<>0
2110   print#2,mid$(str$(v),2);",";
2120   t=(3*t+v) and 63 : if not ef then next k
2130  print#2,not((3*t) and 63);z$;
2140  ll=ll+1
2150  if not ef then next l
2160 gosub 2170 : print#2, t;z$;z$;z$; : close 2
       : close 1 : end
2170 t=ll : print#2,  "00";chr$(ll);z$;d$;:return
```

If information is being prepared for a "collapse" type program, an extra line should be added:

```
2055 get#1,a$,a$
```

Variable D specifies how many items will appear on each DATA line, in this case ten.

After the program containing the DATA statements has been produced, the last line number must be changed to 200 and the generator program itself added.

Summary

Is it worth it? Yes, if a program must be absolutely correct. Yes, if a program is difficult to enter because it contains special characters or needs a machine language monitor.

On the other hand, a "regular" Basic program is much easier – and less work – to enter as a listing. What's more, the reader has a chance to see how it all works.
When you need it, it's handy.

The Transactor
The Tech/News Journal For Commodore Computers

PAYS

$40

per page for articles

We're also looking for professionally drawn cartoons!

Send all material to:

The Editor
The Transactor
500 Steeles Avenue
Milton, Ontario
L9T 3P7

Volume 5 Editorial Schedule

Issue#	Theme	Copy Due	Printed	Release Date
1	Graphics and Sound	Feb 1	Mar 19	April 1
2	The Transition to Machine Code	Apr 1	May 21	June 1
3	Software Protection & Piracy	Jun 1	Jul 23	August 1
4	Business and Education	Aug 1	Sep 17	October 1
5	Hardware and Peripherals	Oct 1	Nov 19	December 1

Advertisers and Authors should have material submitted no later than the 'Copy Due' date to be included with the respective issue.

Driving by Milton?
Come Visit Us!
Take Highway 401 to Highway 25
Go South to Steeles Avenue
(first lights past railway bridge)
Turn left (East) to first lights
100 Yards past there on your right
Is Us!

SEE US AT DAYTON BOOTH 386 & 387

COMMODORE 64™

COMMODORE 64 COLLECTION #1
3D Tic Tac Toe ● 64 Memory Chart ● 64 Renumber ● Baseball 64 ● Black Box ● Checkbook ● Chess for Two ● Concentration ● Connect Four ● Dr. Living ● Dragon Pix ● Entertainer ● Fugue ● Gun Fighter ● Hangman ● Ink Blot ● Joystick Test 64 ● Jumbo Jet Lander ● Labyrinth (J) ● Lander ● Machine Draw ● MeSprite ● Old Shell Game ● Petals ● Program Lister ● Scenarios ● Sprite Ed. Disk ● Sprite Ed. Tape ● Stock Broker ● Trek 64 ● World Time

COMMODORE 64 COLLECTION #2
Hires Disk ● Hires Tape ● Cave Man ● Man ● Pattern ● Sattlite ● Snail ● American Flag ● Baby Care ● Bar Graph ● City Paterns ● Epidemic ● Flight ● Forest Walk ● Geiger Counter ● Life Expectancy ● Master Guess ● Monopole 64 ● Nuke 64 ● O'Hare's Adv. #1 ● Othello for 2 ● Patterns ● Pharoah's Needle ● Poker ● Reg'd Supply ● Supermon 64 ● Temp. Conv.

COMMODORE 64 COLLECTION #3
GAMES — Biorythmn 1.5 ● Checkers ● Clew-So ● Dam Busters ● Hi-Q ● Mille Bourns In ● Mille Bourns PR ● O'Hare's Adv. #2 ● Pig ● Wrap Trap
UTILITIES — Base ● Bio Printer ● Bond Yield ● Calendar ● Dissambler ● FICA Tax ● Food Prices ● Heating Cost ● Intermod ● J/20 Morse R/T ● Menu ● Money Exchange ● PCBN Printer ● Portfolio ● Property Eval ● Timer ● Typing Aid ● Typing Test ● View Disk ● Weight Watcher
EDUCATIONAL — Chemical Drill ● Coll. Like. Terms ● Cryptograms ● ESP Test ● French Sentence ● French Verbs ● Geography ● Grades ● HS Layout ● Integer TTT ● Latin Tutor ● Math Drill ● Metric Quiz

COMMODORE 64 COLLECTION #4
GAMES — Battleship ● Bomardier ● Crops ● General ● Key Move ● Letter Square ● Money Mad ● One Checker ● Osero ● Spades Inst ● Spades Prog ● Stoneville Manor
UTILITIES — Apartments ● Banner ● Big Letters ● Car Cost ● Cash Flow ● Data R/W ● Directory ● Fast Sort ● Global ● Grades ● Interest Rate ● Print Format ● Printer Helper ● Servings ● Sunrise/Sunset ● Weatherman
EDUCATIONAL — Density ● Fractions ● Grammer ● Integer Math ● Long Division ● Mechanics ● Powers & Roots ● Straight Line ● Whole Numbers ● Word Mart

COMMODORE 64 COLLECTION #5
GAMES — Alien Capture ● Auto Rally ● Dominoes ● Explore Lost Gold Adv. ● Niche ● Owari ● superman ● Zone X
UTILITIES — Accounting Test ● Annuity ● Average Growth Building Costs ● Buy/Sell ● Change Disk Name Cross Word ● Data Search ● Elec. Use ● Finance ● Growth Irr. Cash Flow ● Load Calc. ● Real Estate cost ● Rent or Buy
EDUCATIONAL – A or An ● Arrow Facts ● Big Math ● Element Chart Eliza ● Europe ● Hang Math ● Linear Prog. ● Loops Marblestat ● Math Facts ● Number Match ● World Quiz

VIC 20™

VIC-20 COLLECTION #1
2Pet to VIC-20 ● 3 of a Kind ● Addition Game ● Alien 2 ● Alpha Key ● Alphabetizer ● Big Letter Auto ● Big Letters ● Breakout (P) 1 ● Breakout (P) 2 ● Breakout 3 ● Business Demo ● Chuck-A-Luck ● Clock ● Colour Bars ● Commander ● Day of Week ● Decision Maker ● Deflection ● Demo 1 ● Demo 2 ● Demo 3 ● Demo 4 ● Demo 5 ● Demo 6 ● Demo 7 ● Dog Race Inst ● Dog Race Prog ● Dungeon-Game ● Dungeon-Inst ● Etch-A-Sketch ● Happy Birthday ● Hex()Dec ● Home Budget ● Household Invent ● Joystick Test ● Kalidoscope ● Killer Comet ● Kingdom-Game ● Kingdom-Inst ● Lunar Lander ● Lunar Lander 2 ● Math 3 ● Maze Venture ● Merry X-Mas ● Money Changer ● Mukade ● Old English ● Pack Man ● Phone Directory ● Pie Chart (SE) ● Pong Padels ● Premutations ● River Crossing ● Road Race ● Robot Chase ● Rocket Command ● Rocket Command 2 ● Row Boat Song ● Rugby (J) ● Screen Dump ● Silent Night ● Slot Machine ● Snake ● Snowflake Melt ● Squiggle ● Star Wars ● Tank vs UFO ● Tape Finder-VIC ● The Entertainer ● Tiny Tim Inst ● Tiny Tim Monitor ● Word Scrm. Inst ● Word Scrm. Prog

COMMODORE
-USER WRITTEN SOFTWARE-

Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

VIC 20™
collection #1 – collection #2 – collection #3
collection #4 – collection #5 – collection #6
70+ programs per collection – Tape/Disk – $10.00

COMMODORE 64™
64 collection #1 – 64 collection #2 – 64 collection #3
64 collection #4 – 64 collection #5
25+ programs per collection – Tape/Disk – $10.00

PET® / CBM®
5 Utility – Tapes/Disks – $10.00 each
11 Game – Tapes/Disks – $10.00 each
6 Educational – Tapes/Disks – $10.00 each

DINSET™: Reset Switch
Works on Vic 20 or Commodore 64 – $5.00

All prices include shipping and handling.
CHECK, MONEY ORDERS,
VISA and MASTERCARD accepted.

For A Free Catalog Write:

'Public Domain'™, Inc.
5025 S. Rangeline Rd.,
Dept.- T 4.
W. Milton, OH 45383
10:00 a.m. - 5:00 p.m. EST – Mon. thru Fri.
(513) 698-5638 or (513) 339-1725

VIC 20™, CBM™ and Commodore 64™ are Trademarks of Commodore Electronics Ltd.
PET® is a Registered Trademark of Commodore Business Machines, Inc.

VIC-20 COLLECTION #2
23 Matches ● Accounts ● Artillery ● Basic Loader ● Birds Demo ● Blackjack ● Bush Trail ● Calculart ● Camel ● Cards ● Checkbook ● Checkers ● Cirkeldemo ● Colour Wheel ● Craps ● Curs. Letter ● Dam Busters ● Data Search ● Data Sheet ● Dissambler ● Dragon ● Dupe ● Fish Inventory ● Fishing Forecast ● Guess T Number ● Handic ● Histogram ● I Feel Fine ● Installments ● Invest Calc ● Invintro ● Invprog ● Jumble ● Kinetic Art ● Light Show ● Loan Calculator ● Mail List ● Mail List Entry ● Mastermind ● Math Drill ● Math Quiz ● Math Tutor ● Menu Driver ● Monkey ● Moonlander ● Morse ● Nothing Can Go W ● Organ Keyboard ● Othello ● Pacman ● Pacman PT2 ● Patterns ● Piano ● Pinball ● Print My Program ● Programable Char ● Race ● Renumber 1.0 ● Samu ● Sir Duke ● Sound Demo ● Spies ● Super Vicmon ● Tape Counter ● The Muppet Show ● Traffic Signal ● Unit Convert ● Unscramble ● Vicbreak/Padl ● Wall Destroy ● Warfish ● Word Processor

SEE US AT DAYTON BOOTH 386 & 387

VIC-20 COLLECTION #3
37159 Demo ● Acey Ducey Inst ● Acey Ducey Prog ● Address Book ● Air Inductor ● Alpha Key ● Astro-Tron ● Barricade ● Bio Printer ● Brain Teaser ● Color Falls ● Color Squares ● Color Trick ● Color Trick 2 ● Combat ● Contractions ● Countdown ● CW Keyboard ● Days Bet. Dates ● Decibel Calc ● Decision Maker ● Dog Race Inst ● Dog Race Prog ● Dominoes 6K+ ● Dynasquiral ● Exec ● Eye Puller ● Fantansyland ● Flash & Spell ● Flash Cards ● Flash Prompt ● Flight ● Flyer ● Ham Formulas ● Hijacker ● Jingle Bells ● Label Maker ● Lemonade Stand ● Lo Pass Filter ● Look ● Market Inst ● Market Prog ● Math Drill ● Maze ● Mazeman ● Memory ● Phone Directory ● Plow By Numbers ● Poker ● Print Using ● Quad Antenna ● RAM Test ● Reson. Freq. Formul. ● River Crossing ● Screen Fable ● Scroll ● Sequence ● Slot ● Snowflake Melt ● Soc St Rev ● Spelling Challen ● Super Boggle ● Super Demo ● Synonyms-Antonym ● Titration ● Vector Table ● VIC Mail ● Voltage Div ● Word Scrm Inst ● Word Scrm Prog ● Yagi ● Yahtzee

VIC-20 COLLECTION #4
16 Function Demo ● Accountant ● Add Column ● Add Fast ● Bar Graph ● Baseball 6K ● Char-Gen 4D6 ● Charger Inst ● Charger Prog ● Chemical Quiz ● Chief ● Correct Change ● Defuse ● Determinates ● Disk Doctor ● Disk L/R ● E-Brg ● Electronics 1 ● Energy Saver ● Fractions ● French Sentence ● Furnace Calc ● Gross Pay ● Guess-It ● Hangman ● Hi-Res Clock ● Int'l Finance ● Investment F.V. ● Investment File ● J/20 Recv/Trns ● Laser Control ● Log & Logo ● Marquee ● Math Flash ● Matrix Solver ● Memory Monitor ● Missile Math ● Moon Lander ● Mythology ● Noun & Verb Drill ● NPV & Irr-2 ● Pac-Man ● Person Test ● Poker ● Preschool ● Quad-Eq-Solv ● Railroad ● Rotate 8K ● Rounding ● RS232 ● RSR Disable ● Schuifspel ● Sci-Fi Plot Gene ● Simple Scroll ● Sorter ● Speed Reading ● Spelling ● State Capitals ● Std Dev ● Tank Combat ● Tank & Lasers I ● Tank & Lasers P ● Trackmaster ● Triangle Solver ● Tunnel ● TV Test Patterns ● VIC Checkbook ● Weatherman ● Wipeout ● Word Scrambler

VIC-20 COLLECTION #5
Arksnuffel ● Arrows (& Bows) ● Autodelete Lines ● Big Letters 2 ● Biorhythm ● Calendar ● Capitals Intro ● Capitals P1 ● Capitals P2 ● Capitals P3 ● Capitals P4 ● Capitals Quiz ● Cash Flow ● Catalog Maker ● Cave Crawlers I ● Cave Crawlers P ● Change Disk Name ● Classics Quiz ● Concentration ● Decisions ● Dice Roller ● Disk Dissam ● Eater ● English/Metric ● Fallout ● Find Triangle ● Four in a Row ● Gas Milage ● Halyl ● Ham Contest Log ● Hexagon ● Hogan's Heroes ● How Many Boxes ● Keybeep ● Killer Robots ● Life Expectancy ● Long Division ● Loops ● Mathoff ● Mazeman 2 ● Metric/English ● Miauw ● Missing Number ● New Disk ● Notone ● Olympic Theme ● Parts of Speech ● Pinball ● Plays ● Poly Roots ● Polyfit ● Prime # Finder ● Printer Test ● Road Rally ● Robots ● Screen Dump ● Space Trader ● Spades ● Spanish ● Stock Ticker ● Student Grades ● Sub Hunt ● Tank ● Temp Convert ● Typing Drill ● Utility Usage ● Vertical Msg ● VIC Communicator ● VIC Lissajous ● Wall Zap

VIC COLLECTION #6
3D-Maze ● A or An Quiz ● Alarm Quiz ● Alpha Sort Amortize ● Banner ● Big Six ● Bio Printer ● Breakout (J) Budget-Mgr ● Bumblebee ● Buzzword ● Cascades CBuff Peek ● CBuff Save ● Colour Pinball ● Computer Fix Coverage Predict ● Craps ● Crytogram ● Deepspace Density ● Easter Dates ● Electric Cost ● F.I.C.A. Tax Fast Sort ● FFT Calc. ● Firing Tank ● Freq. Coord. Geog. Spell Game ● Guardian ● Hi-Res ● Job Jar L Pad Design ● Labrinth ● Letter Reco ● Marston City Rvd. Match the Number ● Math Dice ● Math Fact Drill Math Test ● Math Time ● Maze Maker ● Merry Christmas Millionaire ● Mosaic ● Multiply Tables ● National Parks Nicoma ● PCB Printer ● PI Network Desig ● Quartic Quiz ● R-S-P ● Rebound Hockey ● Revenge Inst. Revenge Prog. ● Rnd. Graphics ● Rolling Sticks ● Rom Aid Shopping List ● Simon ● Slide Voice ● Snoopy Hires Spelling Aid ● Stopwatch ● Sunrise/Sunset ● Sym T Atten Tennis 3 ● Too High-Too Low ● Typing Test ● Vectors Vehicle Cost ● VIC Orbit ● VIC Symphony ● Wumpus Inst. Wumpus Prog. ● Zero Formula

MICROCOMPUTER SUPPLIES

VISA MasterCard

100% Guaranteed

DISKETTES FROM
10/$25.00
S.S.D.D.

Shop Without Going Shopping

MEMOREX
3481 SS/DD **$33.00 per 10**
3491 DS/DD **$45.00 per 10**
3481-2 SS/DD 2 Pack Mailer
$8.00 plus 75¢ postage

COMMODORE SOFTWARE
COMING SOON

All other sizes and format available and volume discount.

Flip 'N' File, hold 50, 5¼" diskettes	**$38.50 each**
Flip Pac, holds 10 5¼" diskettes	**$ 7.00 each**
Disk Banks, hold 10 5¼" diskettes	**7.00 each**
Plastic Library Cases, choice of colors;	**$4.50 each**
EPSON RX80 Printer	**$475.00**
EPSON FX80 Printer	**$850.00**
EPSON FX100	**$CALL**
MICROWARE Interfaces	**$CALL**
TTX Daisy Wheel Printer	**$775.00**
AMDEK Monitors	**$CALL**
MX 80 Ribbons	**$14.50**

WRITE/TELEPHONE FOR FREE CATALOG.

To order: Send money order, certified cheque, personal cheques must clear our bank, VISA or MASTERCARD. (Include card # and expiry date & signature) Add 5% for shipping and handling, Minimum $3.00 per order. Quebec residents add 9% P.S.T.

INTERNATIONAL MARKETING SERVICES

P.O. Box 522, Boucherville, Quebec, J4B 6Y2
(514)655-9232 Dealer Enquiries Invited

This Space Could Be Transacting For You!

Kelly M. George
Advertising Manager
416 876 4741

INFOMAG INC.

DISCOUNTED PRICE FOR MOST SYST.
APPLE, ATARI, C-64, VIC 20
BRODERBUND *(GAMES)*

Lode Runner D C	$41
Spare Change D	$41
Drol D	$41
Choplifter C D	$41
Seafox D C	$36
(Cartridge version extra)	
Bank Street Writer D	$85

INFOCOM *(ADVENTURES)*

Witness D	$59
Planetfall D	$59

SYNAPSE *(ATARI & C-64, GAMES)*

Fort Apocalypse D T	$41
Blue Max D T	$41

SIRIUS *(GAMES — for most)*

Snake Byte D	$36
Bandits D	$41
Type Attac D	$47
Squish'em C APPLE	$48
D C-64	$41

SMA (SYSTEMS MGT. ASSOC.)

Documate-template C-64	$16
Code pro-64 — Tutorial for basic plus sprite & music gen.	$70

COMM* DATA
EDUCATIONAL (VIC & C-64)

Toddler Tutor	$34
Primary Math Tutor	$34
Math Tutor	$34
English Invaders Games	$34
Gotcha Math Games	$34

Dealer inquiries for:
Programmer's Institute
Kiwisoft
Victory Software
Comm* Data
SMA

C-64, VIC, ATARI, TRS-COLOR PROGRAMMER'S INSTITUTE FUTUREHOUSE)

"Edumate Light Pen"	$36
C-64, VIC, Atari	
"Playground Software" t.m. (Uses Light Pen) C-64 & Atari	
Animal Crackers D	$36
Computer Crayons D	$36
Alphabet Arcade D	$36
Bedtime Stories D	$36

"C.P.A. Complete Personal Account" t.m. C-64, VIC, TRS, Color, Atari

Complete Set (1, 2 & 3) DT	$94
Finance #1 D T	$36
Finance #2 D T	$36
Finance #3 D T	$36
Finance #4 D T	$36

KIWISOFT (C-64)

Paintpic-64	$45
Art on your screen	

VICTORY SOFTWARE
20/64 Dual Packs
Cassettes (T) or Disks (D)

GAMES

Metamorphosis T D	$30
Creators Revenge T D	$30
Labyrinth of Creator T D	$30
Galactic Conquest T D	$30
Kongo Kong T D	$30
Chomper Man T D	$30
Annihilator T D	$30
Adventure Pack I (3 Prog) T D	$30
Adventure Pack II (3 Prog) T D	$30
Bounty Hunter (Adv) T D	$30
Grave Robbers-Graphic (Adv) T D	$24
(Disk version: $4. extra)	

PRECISION SOFTWARE (SILICOM INT'L)

Super Base 64 Data Management System D	$117
Calc Result (Easy)	$108
Calc Result (Advanced)	$202

(C) Cartridge (T) Tape (D) Diskette

Please call for info on your computer model, availability and specific price. Send certified cheque, money order or call and use your visa or Mastercard. Personal cheques require two or three weeks to clear. All prices subject to change without notice. Please include $2.00 per order for postage and handling. Quebec residents only add P.s.T.

Call Toll Free 1-(800)361-0847
except Western Canada, Nfld. and Montreal area (514) call collect
CALL COLLECT (514) 325-6203

between 9 a.m. and 5 p.m. Eastern time
or send order to: 6864 JARRY EAST, MONTREAL, QUE. H1P 3C1

Commodore 64
and
VIC-20

SuperTerm

$149.95

Telecommunications with a difference!

Unexcelled communications power and compatibility, especially for professionals and serious computer users. Look us over; **SuperTerm** isn't just "another" terminal program. Like our famous Terminal-40, **it's the one others will be judged by.**

- **EMULATION**—Most popular terminal protocols: cursor addressing, clear, home, etc.
- **EDITING**—Full-screen editing of Receive Buffer
- **UP/DOWNLOAD FORMATS**—CBM, Xon-Xoff, ACK-NAK, CompuServe, etc.
- **FLEXIBILITY**—Select baud, duplex, parity, stopbits, etc. Even work off-line, then upload to system!
- **DISPLAY MODES**—40 column; 80/132 with side-scrolling
- **FUNCTION KEYS**—8 standard, 52 user-defined
- **BUFFERS**—Receive, Transmit, Program, and Screen
- **PRINTING**—Continuous printing with Smart ASCII interface and parallel printer; buffered printing otherwise
- **DISK SUPPORT**—Directory, Copy, Rename, Scratch

Options are selected by menus and EXEC file. Software on disk with special cartridge module. **Compatible with CBM and HES Automodems**; select ORIG/ANS mode, manual or autodial.

Write for the full story on SuperTerm; or, if you already want that difference, order today!

Requires: Commodore 64 or VIC-20, disk drive or Datasette, and compatible modem. VIC version requires 16K memory expansion. Please specify VIC or 64 when ordering.

Smart ASCII Plus . . . $59.95

The only interface which supports streaming —sending characters simultaneously to the screen and printer — with SuperTerm.

Also great for use with your own programs or most application programs, i.e., word processors. **Print modes:** CBM Graphics (w/many dot-addr printers), TRANSLATE, DaisyTRANSLATE, CBM/True ASCII, and PIPELINE.

Complete with printer cable and manual. On disk or cassette.

VIC 20 and Commodore 64 are trademarks of Commodore Electronics, Ltd.

(816) 333-7200

Send for a free brochure.

MIDWEST MICRO inc.

MAIL ORDER: Add $1.50 shipping and handling ($3.50 for C.O.D.); VISA/Mastercard accepted (card# and exp. date). MO residents add 5.625% sales tax. Foreign orders payable U.S.$, U.S. Bank ONLY; add $5 shp/hndlg.

311 WEST 72nd ST. • KANSAS CITY • MO • 64114

This Space Could Be Transacting For You!

PRO·LINE SOFTWARE

A CANADIAN COMPANY

designing, developing, manufacturing, publishing and distributing microcomputer software

DEALER ENQUIRIES WELCOME
AUTHOR'S SUBMISSIONS INVITED
CALL OR WRITE

(416) 273-6350

PRO·LINE SOFTWARE

755 THE QUEENSWAY EAST, UNIT 8,
MISSISSAUGA, ONTARIO L4Y 4C5

STOCK HELPER™

Commodore 64 and VIC-20

Stock HELPER is a tool to maintain a history of stock prices and market indicators on diskette, to display charts, and to calculate moving averages. Stock HELPER was designed and written by a "weekend investor" for other weekend investors.

Stock HELPER is available on diskette for:

Commodore 64	$30.00	($37.00 Canadian)
VIC-20 (16K)	$27.00	($33.25 Canadian)

plus $1.25 shipping ($1.55 Canadian)

The VIC-20 version only charts 26 bi-weekly periods rather than 52 weekly periods.

(M)agreeable software, inc.

5925 Magnolia Lane • Plymouth, MN 55442

(612) 559-1108

(M)agreeable and HELPER are trademarks of (M)agreeable software, inc. Commodore 64 and VIC-20 are trademarks of Commodore Electronics Ltd.

Intelligent Software for the Commodore computers

Catalog 5/1/83

My line of programs (such as it is) consists of the following products. All are written for Commodore computers; any of my programs will load and run without modification in the entire line (including older PET's).

1. Word Processor; $25. It includes the following features: VERY fast file routines, including a disk file catalog; automatic form handling on tractor- or friction-feed printers; fully imbedded margin, justification, spacing, formatting, and paging controls; block commands and error-trapping in editing mode; and a spool routine (formatted output to disk for later mass printing). I believe W/P is the most thoroughly tested, user-oriented word processor available at this time at anywhere near the price. for any machine. Requires a minimum of 10k of memory (8k expansion on VIC), and a printer.

2. Copycalc; $20 ($15 if ordered with another program). Copycalc is a simplified version of the "electronic spreadsheets" that are becoming extremely popular for use on personal computers. It allows the user to set up a visible grid of numbers on the screen, and use the screen-editor to make changes in the grid, with the totals reflecting the changes. Requires 6k RAM (3k expansion on VIC); smaller version available for unexpanded VIC.

3. Baseball Manager; $30. This program maintains complete batting statistics for a baseball or softball league of up to 250 players. It generates reports on a player, team, or the entire league (including standings). It requires a minimum 10k of RAM; a printer is suggested but not required.

4. Inventory; $30. A general-purpose perpetual inventory control program. It produces a variety of reports, including order forms; multiple vendors are supported. Requires 10k of RAM; a printer is suggested.

All programs: support cassette and disk files and the CBM printers (easily modifiable to other printers), come on cassette, and include documentation. Prices include shipping; Calif. residents add 6%. All programs are copyrighted by the author; those rights will be enforced. Programs available from:

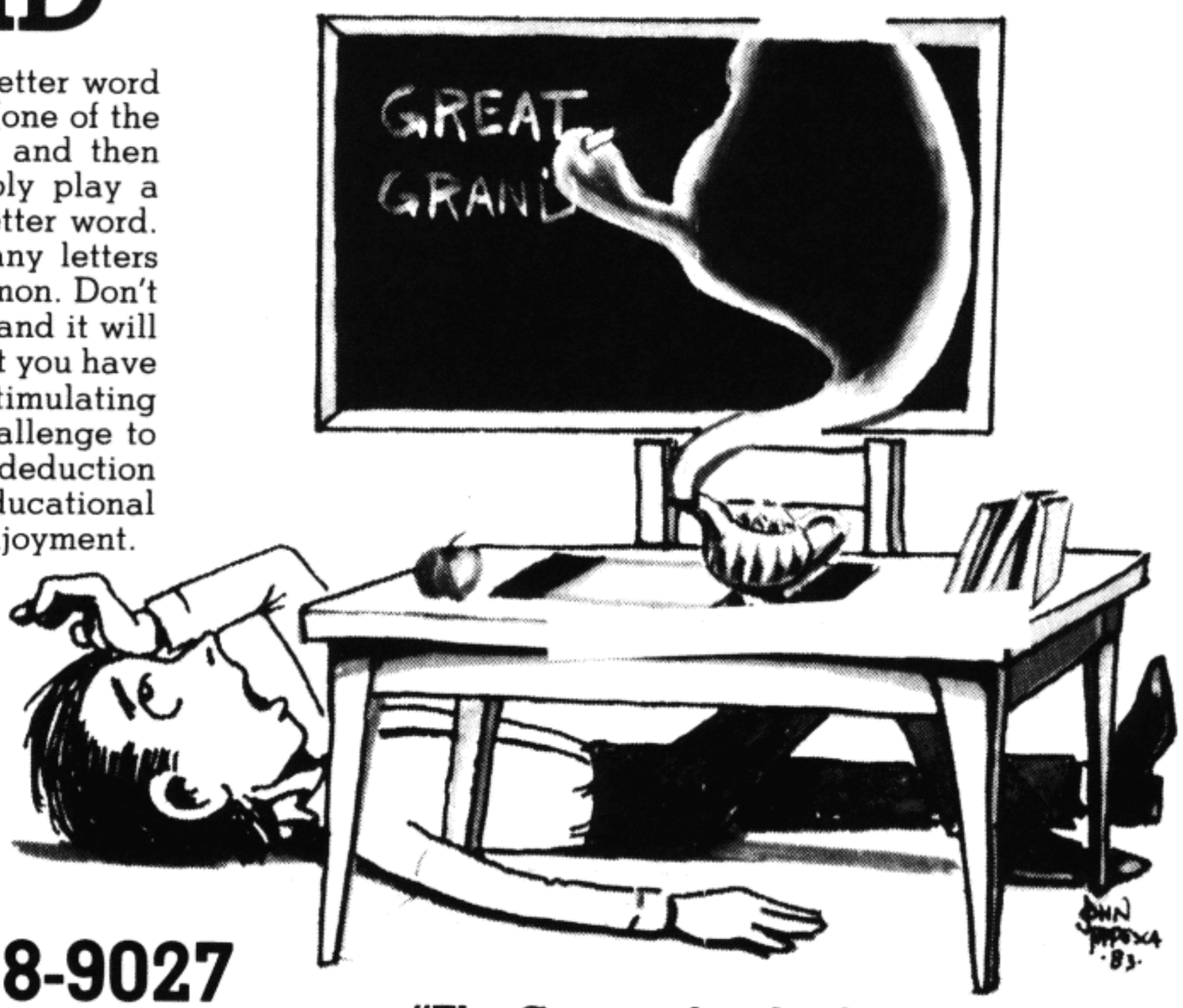
William Robbins, Box 3745, San Rafael, CA 94912

Disk Software for the Commodore 64™

JOT-A-WORD™

A computerized version of the old five letter word game. Simply pick a secret five letter word (one of the almost 5000 words contained on the disk) and then play against the Jot-A-Word Genie or simply play a solitaire version. Start by typing in a five letter word. The Genie responds by telling you how many letters your guess and the secret work have in common. Don't try to cheat, because the Genie is too smart and it will not accept non-words or continue a game that you have given it wrong scores. This is a simple but stimulating game for ages 9 to senior citizen. A real challenge to your intellect, reasoning powers, logic and deduction skills. It's simply hard to beat; as a fun and educational experience! Graphics and music add to the enjoyment.

MasterCard VISA



ONLY $29.95

MICRO WARE 1342B RT. 23 BUTLER, N.J. 07405

Dealers & Distributors Inquiries Invited **201-838-9027**

Prices are in US dollars.

"The Genie is hard to beat!"

WE'LL BACK YOU UP!

ATTENTION COMMODORE 64 OWNERS

If you own a disk drive then you'll need "The Clone Machine". Take control of your 1541 drive.
NEW IMPROVED WITH UNGUARD.*

Package includes:

1.) Complete and thorough users manual
2.) Copy with one or two drives
3.) Investigate and back-up many "PROTECTED" disks
4.) Copy all file types including relative types
5.) Edit and view track/block in Hex or ASCII
6.) Display full contents of directory and print
7.) Change program names, add delete files with single keystroke
8.) Easy disk initialization
9.) Supports up to four drives

MasterCard VISA

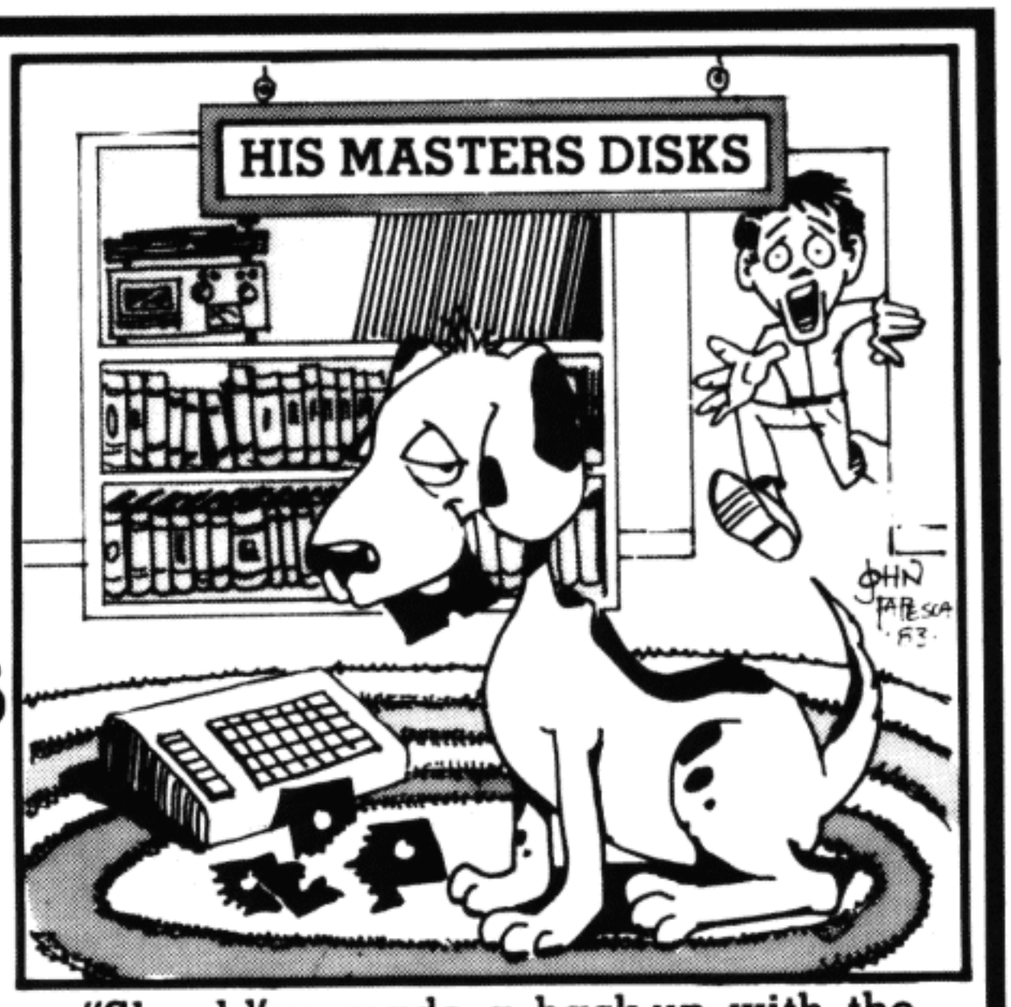
$49.95

*UNGUARD Now allows you to read, write and verify bad sectors and errors on your disk making it easy to back-up most protected software.

Dealers & Distributors
Inquiries Invited

CALL (201) 838-9027

MICRO WARE 1342 B Rt. 23 Butler, N.J. 07405



"Should've made a back-up with the Clone Machine."

DEXTERITY SOFTWARE

*VIC-20

BUG-OUT

A fast action arcade style shooting game. Kill the segments of caterpiller before they kill you. With machine language, for amazing speed. No memory expansion required. $14.95 tape - $19.95 disk

FROGMEN

Another fast action arcade style game. You're a frog trying to get home. You must cross the highway and the river to get five frogs home. With machine language, no expansion required. $14.95 tape
$19.95 disk

LAUNCH

Get your space craft through the asteroid field and out of the galaxy. Machine language for fast action. No Memory expansion required. $14.95 tape
$19.95 disk

*COMODORE 64

VIDEO-PANIC

You are the last human being on earth. You must climb up and down the ladder, avoid aliens and get treasure. Machine language. $14.95 tape $19.95 disk

REBOUND HOCKEY

This is a 100% machine language hockey game for two players. Great color, graphics and realism. Requires two joysticks. $14.95 tape $19.95 disk

WRITE FOR FREE CATALOG....

SEND CANADIAN CHEQUE OR MONEYORDER TO DEXTERITY SOFTWARE P.O. BOX 580 SWIFT CURRENT SASK, S9H 3W4

* VIC 20 AND COMMODORE 64 ARE REGISTERED TRADEMARKS OF COMMODORE BUSINESS MACHINES.

NEW

MPI INTRODUCES SUPER ACTION MEMORY EXPANDER BOARD FOR VIC 20*

Adds 24K and 3 Expansion Slots With Switches and Fuse $109.95

(Expandable To 35K By Simply Adding Memory Chips and Switches)

INTRODUCTORY PRICE

$109.95

Transfer Cartridges To Tape

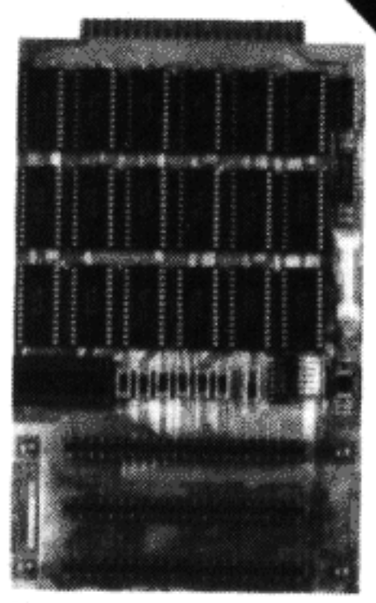
Evade Self Destruct Code With Write Protect Switch

All Boards Factory Tested

Complete Instructions And Illustrations Included.

Fuse Protected

Reset Switch



Pictured Above –
Action Board with all options

24K BOARD FEATURES:

- *Adds 24K Memory (29K with VIC* 5K).
- *Upgrade Board to 35K by adding IC's and switches.
- *Memory switchable in 8K sections. (No need to remove memory board to run your other programs).
- *3 expansion slots with switches (for game or extra utility cartridges).
- *Reset button allows restarting computer without turning power off.
- *.5 amp fuse protected.
- *Switch relocates expansion cartridges in memory so that it can be saved on tape as a backup for your valuable programs. (The unexpanded VIC will not allow cartridges to be saved on tape).
- *Write protect switches allow programs stored in RAM at ROM location or entire Memory to be protected against accidental write.
- *Switch allows memory to be moved between RAM and ROM location. (Useful for developing your own games and saving on tape).
- *Gold plated card edge connector.
- *No other memory expansion needed.
- *Easily plugs into your VIC, no modifications necessary. Saves wear on your VIC 20 since board never needs to be removed or power turned off and on to run other tapes or cartridges.
- **Optional 35K memory (40K with VIC* 5K).

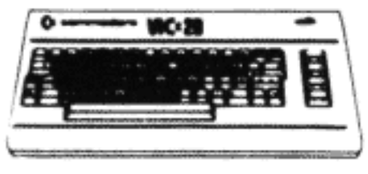
24K memory, 3 expansion slots, switches assembled and tested **$109.95**

Same as above with sockets that allow you to later add your own memory chips to bring memory from 24K to 35K **$124.95**

Full 35K memory, 3 expansion slots, 3K expander mode, eprom socket (switch selectable between BLK 3 & BLK 5) and all switches assembled and tested (eprom not included) **$149.95**

Bare 35K board with complete instruction and parts list **$39.95**

Prices are in US dollars.



VIC 20* is a registered trademark of Commodore.

COD Orders: 816–444–4651

MIDWEST PERIPHERAL INDUSTRIES

MPI

Add $5 for shipping and handling
Mo. residents add 5½% sales tax
Send check or money order to

MPI
Box 8123-B
Kansas City, MO 64112

Personal checks – Allow 3 weeks to clear.

NEW

NEW

NEW

Reduction of an actual sign

The Banner Machine™

For the Commodore 64 (with 5 fonts) or the VIC-20 with 24K memory. • Use on any Epson MX with Graftrax or the FX and RX printers. • Menu-driven program operates like a word processor. • Makes signs up to 10" tall by any length. • Makes borders of variable width up to ¾". • 8 sizes of letters from ¾" to 6½" high. • Proportional spacing; Automatic centering; Right and left justifying. $49.95 Tape or Disk (Specify computer equipment)

For the Commodore 64:

Space Raider An amazing arcade simulation. Your mission is to destroy the enemy ships. $19.95

Super Roller Challenging dice game. Sprite graphics and sound. Yahtzee-style rules of play. $14.95

Formulator A formula scientific calculator designed for tasks which require repetitive arithmetic computations. You can save formulas and numeric expressions. $39.95

Preschool Educational Programs ABC Fun; 123 Fun; and Ginger the Cat with: Addition and Subtraction, Number Hunt, and Letter Hunt. All programs have bright color, music, and action. Each $14.95

Microbroker Exciting, realistic and educational stock market simulation based on plausible financial events. $34.95 Tape or Disk

Sprite Editor The easy way to create, copy, alter, and save up to 224 sprite shapes. $24.95

Cross Reference Generator for BASIC programs Displays line numbers in which any word of BASIC vocabulary appears. Allows you to change variable name and ask for lines where it appears, and more. $19.95

For the VIC-20:

Caves of Windsor A cave adventure game. The object is to restore wealth and happiness to the small village of Windsor. $14.95

Burger & Fries Fast action joystick game. Eat the burgers and fries but avoid the shakes for a top score. $14.95

Catalog available Dealer inquiries invited

PHONE ORDERS: (703) 491-6502
HOURS: 10 a.m. to 4 p.m. Mon.—Sat.

Cardinal Software™

Distributed by
Virginia Micro Systems
13646 Jeff Davis Hwy
Woodbridge, VA 22191

Commodore 64 and VIC-20 are registered trademarks of Commodore Electronics Ltd.

NEW

Assembler for the Commodore 64

PAL64

- **easy to learn**
- **easy to use**
- **fast**
- **comprehensive manual**

Personal assembly language
by Brad Templeton
also available for the Commodore
4,000 – 8,000 – 9,000 series

$99.95 from your local Commodore dealer.

For your nearest dealer call:

(416) 273-6350

PRO·LINE SOFTWARE

755 THE QUEENSWAY EAST, UNIT 8
MISSISSAUGA, ONTARIO L4Y 4C5

PRO GOLF

Here is your chance to play golf on a championship course without all the headaches of getting a tee time, waiting for that slow foursome ahead of you, losing balls, getting rained out or spoiling a good handicap. This game may be played in the privacy of your home or in a clubhouse lounge for the enjoyment of many members. A challenge to even the best players, this game requires a high degree of practice, expertise and accuracy to attain a good score.

PRO GOLF Features :

- A full range of golf clubs (driveway, fairway wood, wedge and irons 2–9)
- Realistic shot distances depending on club and swing
- The ability to hook or slice a shot
- Up to 4 players in one game
- Detailed, colourful screen layouts of 18 different holes (tee, trees, sandtraps, rough, water, out of bounds)
- Simulated ball reaction to course hazards (e.g. ball bounces off trees)
- Hole distances, par, yards to green, strokes taken on hole, total strokes per round and player totals displayed
- A full screen enlargement of greens for putting
- Accurate putting simulation for angle and distance
- Practice of real golf skills – club selection, type of shot (normal, hook, slice), length of swing, special shot strategy (e.g. chipping, getting around or over trees, water, sandtraps)

PRO GOLF
For The Commodore 64
$34.95

written by George Adams

distributed by BMB Compuscience Canada Ltd.
500 Steeles Avenue,
Milton, Ontario
L9T 3P7
416–876–4741

Dealer Inquiries Invited

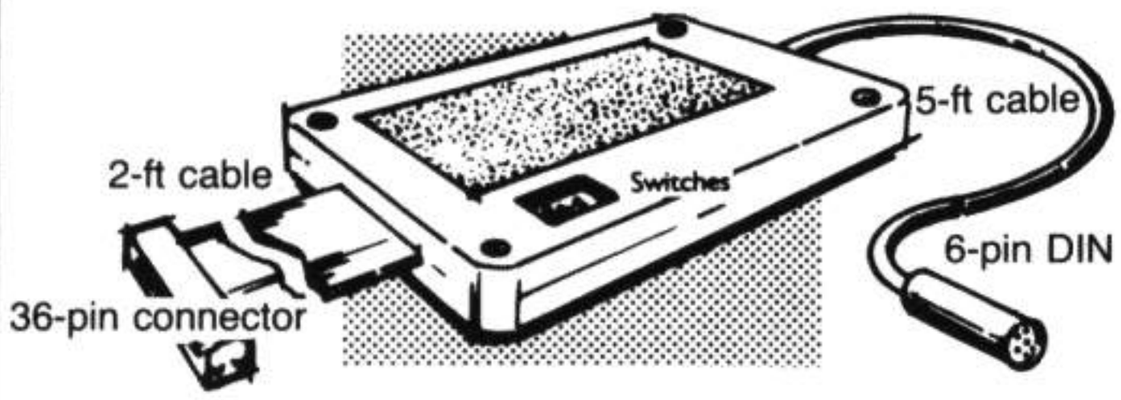
Name _____
Address _____
Prov/State _____ Postal/Zip Code _____

☐Money Order ☐VISA ☐MasterCard ☐Cheque
Acc# _____ Expiry _____
Please include numbers above name

Add $2.00 for shipping & handling
Ontario residents add 7% sales tax.

MW-302: VIC-20/64

Parallel Printer Interface.



Works with all centronics type parallel matrix & letter printers and plotters—Epson, C.Itoh, Okidata, Nec, Gemini 10, TP-I Smith Corona, and most others. Hardware driven; works off the serial port. Quality construction: Steel DIN connectors & Shielded cables. Has these switch selectable options: Device 4, 5, 6 or 7; ASCII or PET ASCII; 7-bit or 8-bit output; upper & lower case or upper only. Recommended by PROFESSIONAL SOFTWARE for WordPro 3 Plus for the 64, and by City Software for PaperClip.

MW-302 . . . . Canadian $189.95

Micro World Electronix, Inc.
3333 S. Wadsworth Blvd. #C105, Lakewood, CO 80227
(303) 987-2671

CANADIAN DEALERS

ALBERTA

Computer Shop of Calgary
3515 18th St. S.W.
Calgary, T2T 4T9
(403) 243-4356

Hindson Computer Systems, Ltd.
7144 Fisher St. S.E.
Calgary, T2H 0W5
(403) 252-9576

TJB Micro Systems, Ltd.
10991 124th St.
Edmonton, T5M 0H9
(403) 433-3161

BRITISH COLUMBIA

Conti Electronics
7204 Main Street
Vancouver, V5X 3Y4
(604) 324-0505

ONTARIO

MGI Computer Corp.
1501 Carling Ave.
Ottawa, T1Z 7M1
(613) 722-1000

Richvale Telecommunications
10610 Bayview (Bayview Plaza)
Richmond Hill, L4C 3N8
(416)884-4165

SASKATCHEWAN

Micro Shack of West Canada
607 45th St. West
Saskatoon, S7L 5W5
(306) 244-6909

COMMODORE OWNERS

Join the world's largest, active Commodore Owners Association.

- **Access to thousands of public domain programs on tape and disk for your Commodore 64, VIC 20 and PET/CBM.**
- **Monthly Club Magazine**
- **Annual Convention**
- **Member Bulletin Board**
- **Local Chapter Meetings**

Send $1.00 for Program Information Catalogue.
(Free with membership).

Membership Fees for 12 Months	**Canada**	**—**	**$20 Can.**
	U.S.A.	**—**	**$20 U.S.**
	Overseas	**—**	**$30 U.S.**

T.P.U.G. Inc.
Department "M"
1912A Avenue Road, Suite 1
Toronto, Ontario, Canada M5M 4A1

*** LET US KNOW WHICH MACHINE YOU USE ***

NEW

Basic Utility for the Commodore 64

POWER 64

•easy to learn
•easy to use
•program faster and more efficiently with better results
•MOREPOWER included FREE

Powerful Programmer's Utility
by Brad Templeton
Manual by Jim Butterfield

$99.95 from your local Commodore dealer.
For your nearest dealer call:
(416) 273-6350
PRO·LINE SOFTWARE
755 THE QUEENSWAY EAST, UNIT 8
MISSISSAUGA, ONTARIO L4Y 4C5

MAILPRO
STEVE PUNTER'S DATA ORGANIZER AND MAILING LIST PROGRAM FOR COMMODORE 64™

COMPARE THESE FEATURES:

- **fast file definition**
- **easy updating**
- **rapid printing with total format and record selection control**
- **WORDPRO compatible**
- **up to 4000 records on 1541**

MAILPRO 64.....$129.95

Also available for COMMODORE 8032 . . . $179.95

Call for the name of your local dealer:

PRO·LINE SOFTWARE

PRO-LINE SOFTWARE LTD.

(416) 273-6350

755 THE QUEENSWAY EAST, UNIT 8
MISSISSAUGA, ONTARIO CANADA, L4Y 4C5

C 64 PROVINCIAL PAYROLL

A complete Canadian Payroll System for Small Business.

• 50 Employees per disk (1541) • Calculate and Print Journals • Print Cheques • Calculate submissions summary for Revenue Canada • Accumulates data and prints T-4s • Also available for 4032 and 8032 Commodore Computers.

Available from your Commodore Dealer.

Distributed by:

MICROCOMPUTER SOLUTIONS

1262 DON MILLS RD. STE. 4
DON MILLS, ONTARIO M3B 2W7
TEL: (416) 447-4811

COMMODORE COMPUTER PRINTER ADAPTERS

—addressable-switch selectable upper/lower, lower/upper case.
—works with BASIC, WORDPRO, VISICALC and other software.
—IEEE card edge connector for connecting disks and other peripherals to the PET.
—power from printer unless otherwise noted.

RS-232 SERIAL ADAPTER—
baud rates to 9600 — power supply included.
MODEL ADA 1450a $149.00

CENTRONICS/NEC PARALLEL ADAPTER — Centronics 36 pin ribbon connector — handles graphics.
MODEL ADA 1800 . . . . . $129.00

COMMUNICATIONS ADAPTER-
serial & parallel ports — true ASCII conversion — baud rates to 9600 — half or full duplex — X-ON, X-OFF - selectable carriage return delay — 32 character buffer — centronics compatible.
MODEL SADI $295.00

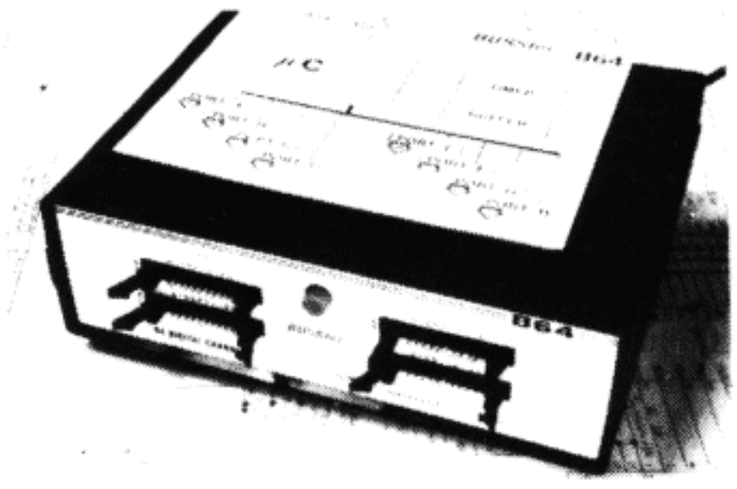
COMMODORE 64 to RS-232 CABLE ADAPTER
MODEL ADA 6410 . . . . . $79.00

Prices are in US dollars.

BUSSter COMPUTER INTERFACES

IEEE — 488
RS — 232

ANALOG AND DIGITAL INPUT/OUTPUT MODULES



The BUSSter line of analog and digital products was designed to collect data and to output signals to laboratory and industrial equipment in conjunction with a microcomputer system. These powerful self-contained modules reduce a computer's workload by providing read or write operations to external devices. They are controlled as slave interfaces to real-world physical applications. Control is over an IEEE-488 (GPIB) bus or RS-232 port. BUSSter modules are available in several digital and analog configurations. The internal buffer and timer provide flexibility by allowing the BUSSter to collect data while the host computer is busy with other tasks.

BUSSter A64—64 channel digital input module to read 64 digital signals. Built-in buffer **$495.00**

BUSSter B64—64 channel digital output module to send 64 digital signals **$495.00**

BUSSter C64—64 channel digital input/output module to read 32 and write 32 digital signals. Built-in buffer **$495.00**

BUSSter D16—16 channel analog input module to read up to 16 analog signals with 8 bit resolution (¼%) Built-in buffer **$495.00**

BUSSter D32—32 channel version of the D16 **$595.00**

BUSSter E4—4 channel analog output module to send 4 analog signals with 12 bit resolution (.06%) **$495.00**

BUSSter E8—8 channel version of the E4 **$595.00**

BUSSter E16—16 channel version of the E4 **$695.00**

Add the suffix -G for IEEE-488 (GPIB) or -R for RS-232.

All prices are USA only. Prices and specifications subject to change without notice.

30 DAY TRIAL—
Purchase a BUSSter product, use it, and if you are not completely satisfied, return it within 30 days and receive a full refund.

US Dollars Quoted
$10.00 Shipping & Handling
MASTERCARD/VISA

CmC

Connecticut microComputer, Inc.
INSTRUMENT DIVISION
36 Del Mar Drive
Brookfield, Ct. 06804
(203) 775-4595 TWX: 710-456-0052

SOFTWARE FOR
VIC ★ COMMODORE 64 ★ PET
FROM KING MICROWARE

- S D COPY FAST EFFICIENT SINGLE DISC COPIER FOR THE 1541 $19.95
- WORDS & CALCS SPREAD SHEET FOR THE C-64 ALLOWS TEXT $42.95
- CHART PAC 64 FINEST CHART MAKER AROUND $42.95
- SMARTEES ACTION PACKED MAZE GAME $22.95

NEW! THE BANKER THE FINEST CHECK BOOK RECONCILIATION PROGRAM ON THE MARKET $38.95

- DAISY — DATA ADAPTABLE INFORMATION SYSTEM
  — THE DATA BASE WITH A DIFFERENCE
  — ALLOWS YOU TO CALCULATE BETWEEN FIELDS $39.95
- ASTRO POSITIONS FIND THE STARS AND CAST YOUR HOROSCOPE $43.95

LOOK AT THE LANGUAGES WE HAVE

YES! WE HAVE PASCAL $52.95

ULTRABASIC WITH TURTLE GRAPHICS AND SOUND $42.95

TINY BASIC COMPILER $22.95

TINY FORTH FIG FORTH IMPLEMENTATION $22.95

EDIT/ASM COMPLETE EDITOR ASSEMBLER PACKAGE $36.95

64-BUDGETEER
64-CRIBBAGE
SKIER-64
64 QUICK-CHART
SYNTHY-64

VIC TINY PILOT
VIC BUDGETEER
VIC VIGIL
VIC CRIBBAGE
GRAPHVICS

SCREEN DUMP
SPRITE-AID
VIC HIRES
VIC JOYSTICK PAINTER
VIC I-CHING

We are actively seeking SOFTWARE AUTHORS.
WHY NOT SEND US YOUR PROGRAM FOR EVALUATION.

Dealer Inquiries Invited
Write for our FREE Catalogue
for VIC and C-64

Canadian manufacturer and distributor for ABACUS Software Products

KING MICROWARE Ltd.

Suite 210,
5950 Côte des Neiges
Montreal, Quebec H3S 1Z6

SuperPET USERS!

Have you ever wondered what the /%*%$%& is the difference between 'c*/ %*//' and '*c/ %*//'? Tired of flipping that switch just to do a 'COLLECT'? The SUPERPET TUTORIAL DISK reveals the mysteries of the data editing commands and 'meta-character' strings, using clear and useful examples. It also contains:

* Syntax of all microEDITOR search strings.
* Information on all SuperPET/PET file types.
* Examples of all the disk access commands.
* Instructions on issuing all DOS commands.
* Information on RS-232C and terminal facilities.
* A table of system addresses and switches.
* Decimal & hex addresses of all 6809 ROM routines.
* Hexadecimal and decimal ASCII translation tables.

A general description of the microEDITOR.
Explanations of ALL microEDITOR commands.
Explanations of ALL microMONITOR commands.
Explanations of all SETUP menu options.
Explanations of all DOS error messages.
Information on the Programmed Function Keys.
Instructions on automating disk maintenance.
Hexadecimal-decimal conversion table.
6809 Assembler opcodes, modes and lengths.

This product costs only $39.95, postage and handling included. The items marked with '*' are also available on a Reference Card (which is included free with each Tutorial Disk ordered). The Reference card alone costs just $10. If you order the Tutorial, the disk you receive will also contain a selection of the best public-domain SuperPET software from various sources. Also available is the APL-microEDITOR interface, which allows use of the microEDITOR for editing APL functions and variables. Volume discounts are available (40% off for 2–10 50% off for 11–100).

Send a check now (and specify 4040 or 8050 format); or write for more information (including a free SuperPET bibliography) to:

Dyadic Resources Corporation
P.O. Box 1524 Station 'A'
Vancouver, B.C. Canada V6C 2P7

('c*/ %*//' hangs up; '*c/ %*//' does nothing; but '*c*/ %*// left justifies.)

COMMODORE 64™ COMAL ADDS:

- 40 Graphics Statements
- 10 Sprite Statements
- "LOGO" TURTLE GRAPHICS
- RUN-TIME COMPILER
- FAST program execution
- auto line numbering
- line renumbering
- program structures
- merging program segments
- long variable names
- named procedures
- parameter passing
- local and global variables
- random access disk files
- stop key disable
- End Of File detection

What does this and more? **COMAL**
What is the cost? **Only $19.95**

All this and much, much more on disk with many sample programs. ONLY $19.95. Also available: COMAL HANDBOOK, $18.95. BEGINNING COMAL, $19.95. STRUCTURED PROGRAMMING WITH COMAL, $24.95. FOUNDATIONS IN COMPUTER STUDIES WITH COMAL, $19.95. CAPTAIN COMAL GETS ORGANIZED, $19.95. COMAL TODAY newsletter, $14.95. Send check or Money Order in US Dollars plus $2 handling to: COMAL Users Group, U.S.A., Limited, 5501 Groveland Ter., Madison, WI 53716 phone: 608-222-4432. COMMODORE 64 is trademark of Commodore Electronics Ltd. CAPTAIN COMAL is trademark of COMAL Users Group, U.S.A., Limited.

CASSETTES ! ! !

C-10 For 35c (100 LOT)

FOR YOUR COMPUTER

ALL CASSETTES ARE:

- DIGITAL COMPUTER GRADE TAPE
- HIGH FREQUENCY RESPONSE
- WIDE DYNAMIC RANGE
- 100% ERROR FREE
- 5 SCREW HOUSING
- FULLY GUARANTEED
- CAREFULLY PACKED

—All Prices Include U.S. Shipping and Handling—
—PHONE ORDERS ADD $2.50 C.O.D. FEE—
[*FOR CANADIAN ORDERS SEE BOTTOM OF AD*]

**"COMPUTER GRADE" tape is very high response oxide formulation by a Major U.S. Manufacturer.
**BASF tapes are the top of the line oxide formulation for the BASF corporation.
**For orders of 1000 or more you may specify color choice (Black, Beige, or White).

****

— CUSTOM LENGTHS AVAILABLE —
COMPUTER TAPE PRICES

LENGTH	25 LOT	100 LOT	1000 LOT
	COMPUTER GRADE TAPES		
C-5	.45/11.25	.35/35.00	.30/300.00
C-10	.50/12.50	.35/35.00	.30/300.00
C-20	.50/13.75	.40/40.00	.35/350.00
C-30	.60/15.00	.45/45.00	.40/400.00
	— BASF DPS GRADE TAPES —		
C-5	.50/12.50	.40/40.00	.35/350.00
C-10	.55/13.75	.40/40.00	.35/350.00
C-20	.60/15.00	.45/45.00	.40/400.00
C-30	.65/16.25	.50/50.00	.45/450.00

5000 Lot Prices . . . . . deduct 1c from 1000 lot price
10000 Lot Prices . . . . . deduct 2c from 1000 lot price

****

Quality Norelco Cassette Case...and Label Prices

[*WITH CASSETTE ORDERS ONLY*]

25-249 Cases/.20 Ea.	250/.13 Ea.		1000/.11 Ea.
25-499 Soft Poly Cases/.18 Ea.			500/.11 Ea.
Labels — Sheet	12/.20	120/1.70	1200/14.50
Tractor Feed Cassette Labels (1 up)			1000/14.50

High Speed Digital Computer Cassette Duplicators
(call for information)

Send Cashier's Checks, Money Orders, and Checks To:

CASS-A-TAPES
Box 8123-T
Kansas City, Mo. 64112
816-444-4651

For Canadian Orders ONLY add shipping as follows:
25-99 tapes add $2.00 to total cost
For each 100 tapes ordered add $6.00 to total cost
For cases add same amounts as above for shipping

Let The SMART 64 Terminal

COMMODORE 64* Do The DRIVING

No matter which direction you wish to travel in, experience the advantage of computer communications with The SMART 64 Terminal. Discover the program that puts you on the Right Road to: Public-Access Networks, University Systems, Private Company Computers and Financial Services.

The SMART 64 Terminal designed with Quality-Bred features, Affordable Pricing . . . And Service.

So why not travel the communications highways the SMART way!

Suggested **$39.95** Retail

Accessories included:

- ☐ *Selective Storage of Received Data.*
- ☐ *Alarm Timer.*
- ☐ *40 or 80 Col. Operation*.*
- ☐ *Auto-Dial.*
- ☐ *User-Defined Function Keys, Screen Colors, Printer and Modem Setting.*
- ☐ *Screen Print.*
- ☐ *Disk Wedge Built-In!*
- ☐ *Formatted Lines.*
- ☐ *Review, Rearrange, Print Files.*
- ☐ *Sends/Receives Programs and Files of ANY SIZE.*
- ☐ *Adjustable transmit/receive tables allow custom requirements. These and other features make The SMART 64 Terminal the best choice for grand touring telecommunications.*

*Commodore 64 registered trademark of Commodore Business Machines Inc.

*Supports 80-column cartridge by Data 20 Corporation.

Dealer Availability
Call (203) 389-8383

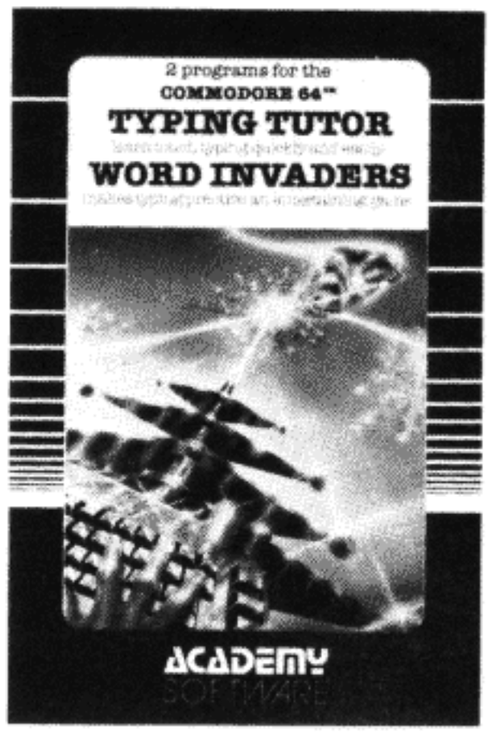
MICROTECHNIC® SOLUTIONS
P.O. BOX 2940, New Haven, Ct. 06515

Prices are in US dollars.

VIC 20™ COMMODORE 64™

JOIN THE COMPUTER REVOLUTION WITH A MASTERY OF THE KEYBOARD!

In the age of the computer, *everyone* from the school child to the Chairman of the Board should be at home at the computer keyboard. Soon there will be a computer terminal on every desk and in every home. Learn how to use it right ...and have some fun at the same time!



Rated THE BEST educational program for the VIC 20™ by Creative Computing Magazine

TYPING TUTOR PLUS WORD INVADERS

The proven way to learn touch typing.

COMMODORE 64 Tape $21.95 COMMODORE 64 Disk $24.95
VIC 20 (unexpanded) Tape $21.95

Typing Tutor plus Word Invaders makes learning the keyboard easy and fun! Typing Tutor teaches the keyboard in easy steps. Word Invaders makes typing practice an entertaining game. Highly praised by customers:

"Typing Tutor is great!", "Fantastic", "Excellent", High quality", "Our children (ages 7-15) literally wait in line to use it.", "Even my little sister likes it", "Word Invaders is sensational!"

Customer comment says it all . . .

". . . it was everything you advertised it would be. In three weeks, my 13 year old son, who had never typed before, was typing 35 w.p.m. I had improved my typing speed 15 w.p.m. and my husband was able to keep up with his college typing class by practicing at home."

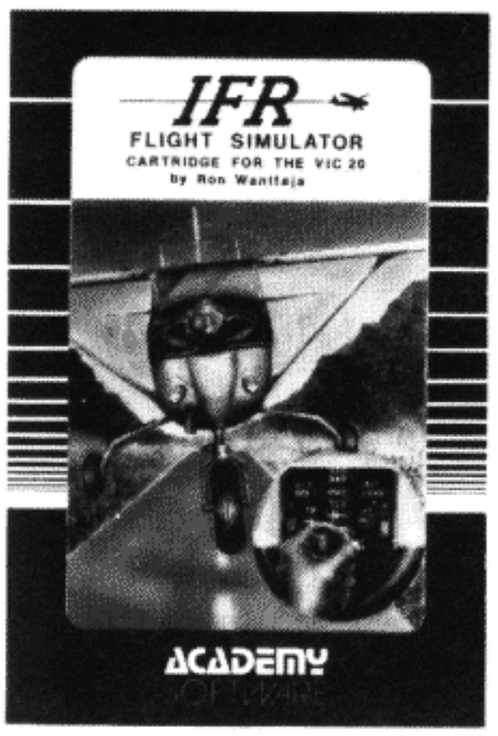
NEW!

IFR (FLIGHT SIMULATOR) CARTRIDGE FOR THE VIC 20

$39.95

COMMODORE 64 TAPE OR DISC $29.95

JOYSTICK REQUIRED



Put yourself in the pilot's seat! A very challenging realistic simulation of instrument flying in a light plane. Take off, navigate over difficult terrain, and land at one of the 4 airports. Artificial horizon, ILS, and other working instruments on screen. Full aircraft features. Realistic aircraft performance — stalls/spins, etc. Transport yourself to a real-time adventure in the sky. Flight tested by professional pilots and judged "terrific"!

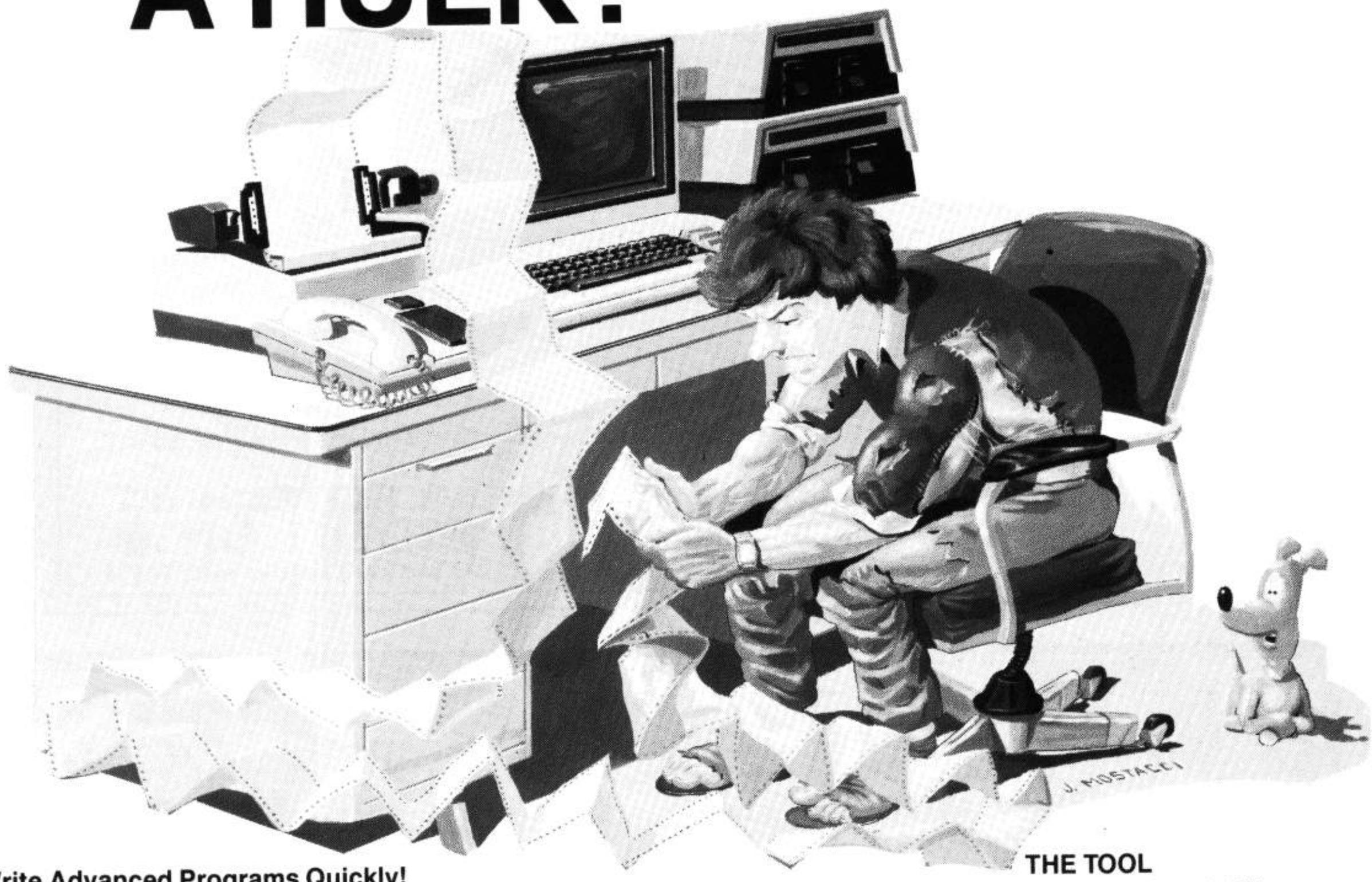
VISA Shipping and handling $1.00 per order. CA residents add 6% tax. MasterCard

ACADEMY SOFTWARE

P.O. Box 9403, San Rafael, CA 94912 (415) 499-0850

Programmers: Write to our New Program Manager concerning any exceptional VIC 20™ or Commodore 64™ game or other program you have developed.

IS PROGRAMMING TURNING YOU INTO A HULK?



Write Advanced Programs Quickly!

Tired of writing reams of code? Take a quantum jump into the future! Tommorrow's programmers are using software development tools such as THE TOOL. THE TOOL lets you make use of powerful machine language subroutines. Your programs will execute fast using less code. Input/output routines and professional looking screens are easily created.

Features of **THE TOOL** include :

- Screen Design functions which allow controlled input and output
- High Resolution Graphics with alpha/numeric display
- Screen Save and Load functions (for hi-res and text screens)
- Structured BASIC instructions , e.g. IF THEN ELSE
- Programming Aids (e.g. auto, renumber, delete, find, trace, hardcopy)
- 2 keystroke disk commands (DOS support extensions)
- Game Design Instructions (joy, scroll, screen, colour)
- A 50 page user manual

THE TOOL
For The Commodore 64 ™
$65.00

developed by Micro Application

distributed by BMB Compuscience Canada Ltd.
500 Steeles Avenue,
Milton, Ontario
L9T 3P7
416–876–4741

Dealer Inquiries Invited

Name _____

Address _____

Prov/State _____ Postal/Zip Code _____

Order ☐VISA ☐MasterCard ☐Cheque

Acc# _____ Expiry _____

Please include numbers above name

Add $2.00 for shipping & handling
Ontario residents add 7% sales tax.

CANADIAN SOFTWARE SOURCE

COMPUTERWARE CATALOGUE SHOPPING

GREAT SAVINGS
FAST*
FRIENDLY
SERVICE

► VISIT US AT THE PERSONAL AND BUSINESS COMPUTER SHOWPLACE, APRIL 6 to 8/84 C.N.E. TORONTO

OVER 700 COMMODORE 64 PRODUCTS

CANADIAN DOLLARS	Retail	C.S.S.		Retail	C.S.S.
HOME ACCOUNTANT (Continental) (D)	$ 99.95	$ 74.95	WEIGHT CONTROL (Simutech) (CAR)	$ 69.95	$ 58.95
BANK STREET (BRODERBUND) (D)	99.95	69.95	BEAR JAM (Chalkboard) (CAR)	49.95	42.95
PAINT MAGIC (Datamost) (C & D)	69.96	55.95	EARLY GAMES FOR YOUNG CHILDREN (Cntpt) (T & D)	39.95	34.95
COMBAT LEADER (SSI) (D)	49.95	44.95	GOTCHA MATH GAMES (Comm*Data) (D)	Call	Call
KOALA PAD (Koala Tech)	149.95	107.50	MATH TUTOR (Comm*Data) (D)	Call	Call
KEN USTON'S BLACKJACK (Screenplay) (D)	94.95	82.50	SPELLOCOPTER (Designware) (D)	49.95	44.95
MINER 2049'er (RESTON) (CAR)	59.95	49.95	TRIVIA 1 (Cymbal) (D)	49.95	44.95
STARFIRE/FIRE ONE (Epyx) (T & D)	49.95	39.95	SYSRES (Solidus) (D)	115.95	99.95
WAY OUT (Sirius) (D)	51.95	45.95	PRINTER INTERFACE +G (Cardco)	149.95	119.95
FAMILY TREE GENEALOGY (D)	49.95	46.95	GEMINI - X (Star Miconics) including PRINTER INTERFACE +G (Cardco)	Special	495.00
FCM (Continental)	69.95	54.95	BEACH-HEAD (ACCESS) (T & D)	69.95	37.95
THE HYPNOTIST (Psycom) (D)	149.95	139.95	IN SEARCH OF... (Spinnaker) (D)	52.95	42.95
FLIGHT SIMULATOR II (Sublogic) (D)	69.95	59.95	PRINTER PAPER - 30 M - 9½" x 11" (3300 shts/box)	Special	42.95
STRIP POKER (Artworx) (D)	54.95	48.95	MUSIC CALC (Wave form) (D)	100.00	75.00
ASTROPOSITIONS (King Microware) (D)	49.95	43.95	MUSIC CONSTRUCTION SET (Electronic Arts) (D)	59.95	53.95
PAPER CLIP (Batteries included)	**IBC	99.95	JOYSTICK (Kraft)	Special	25.00
FRENCH EASYSCRIPT 64 (Silicom Int.) (D)	**IBC	129.95			
SEXUAL CONFIDENCE (Simutech) (CAR)	69.95	58.95			

*IBC — INCLUDES 10% BONUS CREDIT

FOR ORDER OR FREE CATALOGUE WRITE OR PHONE CANADIAN SOFTWARE SOURCE

BOX "340" STATION "W", TORONTO, ONTARIO M6M 5B9 (416) 491-2942

VISA

Ontario Residents add 7% sales tax. Send certified cheque or money order. Visa and Master Card please include card number, expiry date and signature.

Add $2.50 for shipping and handling. All items subject to availability. Prices subject to change without notice.

*Delivery by U.P.S. within 3 days of order date if stocked by local suppliers.

MasterCard

For affordable business software, put MICROSPEC in MEMORY

MicroSpec is serious about offering sophisticated business programs at an affordable price. And there's no skimping on the quality or capabilities of the programs.

Whether you're using **DBM, INV, G/L, PAY, A/R, A/P, Fixed Asset, Rental Property Management, Mail List, or Checkbook Manager** you'll find MicroSpec programs offer the greatest flexibility and optimum performance for your specific needs. And you'll like the price!

Proven MicroSpec business programs are available NOW for the Commodore 64 and VIC 20 (trademarks of Commodore). Applications for other popular micros will be coming soon.

A leader in the design of affordable business systems – MicroSpec. Put it in your memory.

For more information, contact your local dealer.

MicroSpec, Inc.
P.O. Box 863085 • Plano, TX 75086
(214) 867-1333

(Dealer inquiries welcome)

MICROSPEC

COMMODORE SOFTWARE AFICIONADOS

You'd like free software, we'd like reviews. The Book Company seeks additional reviewers for *The Book of Commodore Software*. For details, write and send a sample review to:

The Book Company
11223 S. Hindry Ave.
Los Angeles, CA 90045

Advertising Index

Software

Advertiser	Issue# / Page 2	3	4	5	6	1	Product Name (Description)	Manufacturer
Academy Software			73	77	78	**89**	VIC20/C64 Software	
Beacon Software		74		85			Business Packages	
BMB Compuscience				73	72	**90**	The TOOL (programming aid)	Micro Application
				76	68	**84**	Pro Golf	
Boston Educational Computing			64				Educational Software	
Canadian Software Source					79	**91**	C64 Software	
Cass-A-Tapes						**88**	Commodore software	
Cardinal Software				72	62	**83**	VIC20/C64 Games, Utilities, Edu.	
COMAL Users Group					65	**88**	C64 COMAL	
Computer Marketing\Canadian Micro	IBC	IBC	IBC				Calc Result (spreadsheet prog.)	Handic Software ab
Dexterity Software						**82**	C64/VIC 20 games	
Dyadic Resources Corp.						**88**	SuperPET information	
Eastern House			68	81	71		MAE Assembler	
Execom Corp.			68				40/80 Screen Select	
Hofacker				87			C64 wordprocessor	
Info Mag Inc.						**79**	Commodore software	
Input Systems Inc.		68	81	71			Typro (wordprocessor)	
Isis Hathor			IBC	IBC	IBC	**IBC**	Laser Strike	
King Microware				76		**87**	VIC/64/PET software	
Magreeable Software		66	75	74		**81**	Stock Helper	
Microcomputer Solutions			81	71		**86**	C64 Provincial Payroll	
Micro Ware			75	74		**81**	C64 JOT-A-WORD	
			82	65		**82**	C64 Disk Utility	
MicroSpec			70	69		**91**	C64/VIC 20 Business Software	
Microtechnic Solutions			77	78		**89**	C64 Terminal software	
Midwest Micro Inc.		69					VIC20/C64 Graphics Util	
			83	73		**80**	VIC20/C64 SuperTerm	
Performance Micro Products		64		65			C64 Forth	
P.F. Communications			86				J Butterfield video tutor	
Pro–Line Software	79	73	72	62		**2**	Commodore software	
	79	73	72	62		**83**	PAL 64 (assembler)	
	76	69	74	77		**85**	POWER 64 (programming aid)	
		74	78	64		**86**	MailPro	
	76	66	83	73		**80**	general	
Psycom Software Int'l			80	79			C64 software	
Public Domain Inc.						**78**	Commodore software	
RAK Electronics				80			VIC20/C64 Games, Utilities	
Silicom International					75		SuperBase 64 (data base)	Precision Software
Southern Solutions		78	71				Business packages	
William Robbins Software			74	75	74	**81**	VIC/64/PET Software	

Hardware

Advertiser	Issue# / Page 2	3	4	5	6	1	Product Name (Description)	Manufacturer
Apropos Technology				79	67		VIC20/C64 Printer, Exp board	
Computer Workshops \Computer Marketing	63						Z–RAM (CP/M board)	Madison
Connecticut microComputer	62	79					PET/CBM Interface adapters	
			64	78	64	**86**	Analog/Digital I/O	
Eastern House			68				Trap 65	
			68				VIC Rabbit	
			68				Eprom Programmer	
			68				Communications Bd	
George M. Drake & Associates			BC	BC	BC	**BC**	Colour Monitors	Amdek
Gosub International			75	82	63		Flexikey	
Micro Ware			70				Tape Interface	
			70				VIC20 RAM Expand	
Micro World Electronix			65	74	77	**85**	VIC20/C64 Printer Interface	
Midwest Micro Inc.			69				Smart ASCII Plus	
Midwest Peripherals			74	72	62	**83**	VIC20 Expander	
Precision Technology	61	79	73				VIC20/C64 Expander Boards	
Richvale Telecommunications	IFC	IFC	IFC	IFC	IFC	**IFC**	C64 Link (IEEE adapter) + software	
Zanim Systems				84	70		Home control hardware	

Accessories

Advertiser	Issue# / Page 2	3	4	5	6	1	Product Name (Description)	Manufacturer
The Book Company						**91**	Software review/exchange	
Computer Workshops		76					Apr83 products list	
The Code Works			69	78	64		'CURSOR', C64 Tape Magazine	
Int'l Marketing Services				80	79	**79**	Disk, printers, misc.	
Midnight Software Gazette			72	71	66		Subscriber Info	
Toronto PET Users Group	61		75	74	77	**85**	Membership info	
Zanim Systems						**1**	CAD/CAM Tutorial	

Laser Strike

for the commodore 64

challenge the asteroid field,
maneuver the caves of ice,
experience the thrill,
play laser strike.

Laser strike, written in full machine language for the Commodore 64.

Commodore 64 is a registered trademark of Commodore Business Machines Inc.

Visa/MC/Check/Money Order accepted

In U.S.
Cassette $24.95
Disk $29.95
Isis Hathor Digital Productions
6184 Verdura Ave.
Goleta, CA 93117
(805) 964-6335
Add $2.00 postage and handling
California residents add 6% sales tax

*** Ask about Laser strike posters**

Call Toll Free - 1-800-558-8803

ISIS HATHOR
DIGITAL PRODUCTIONS

In U.K.
Cassette £ 9.00 VAT included
Disk £19.95 VAT included
Isis Hathor U.K.
Andrew Barrow
Royden, Perkslane
Prestwood, Gt. Missenden
Bucks, England HP16 OJD
02406-3224
You will be billed for postage and handling

COMPATIBLE COLOR-I . . .

NEW 2 YEAR WARRANTY!
On all monitor electronics . . . 3 yrs. on all CRT's
(See details at dealer)

The popular choice for popular computers . . . at a popular price.

The Color-I Monitor is designed to perform superbly with your Apple II, Atari or VIC Commodore personal computer and others. Highly styled cabinet. It accepts a composite video signal to produce vivid, richly colored graphic and sharp text displays. Very reasonably priced, the Color-I is a giant step above home TV sets and other monitors.

Just write, or call to receive complete specifications on the Amdek Color-I Monitor.

- Quality 260(H) x 300(V) line resolution.
- Built-in speaker and audio amplifier.
- Front mounted controls for easy adjustment.
- Interface cables available for Atari and VIC Commodore computers.
- FCC/UL approved.

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786
REGIONAL OFFICES: Calif. (714) 662-3949 • Texas (817) 498-2334

AMDEK CORP.

Amdek . . . your guide to innovative computing!